DAD-Less MIPv6, an improved mechanism for MIPv6

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science

in the University of Canterbury

by

Yu-xuan (Tim) Hong

Supervisory Committee:

Prof Krzysztof Pawlikowski	Supervisor
Prof Harsha Sirisena	Co-Supervisor
Dr Allan McInnes	Associate Supervisor

Examining Committee:

Prof. P. Komisarczuk, Thames Valley University, London, UK

Prof. K. Pawlikowski, University of Canterbury, Christchurch, NZ

University of Canterbury

2010

This thesis is dedicated to my parents for their love and support.
I love you both deeply.

Abstract

Due to the exhaustion of IPv4 addresses, the deployment of IPv6 has become imminent ever. With the increasing popularity of mobile wireless connections, a comprehensive mobile IPv6 protocol is high demanded. However, the current MIPv6 standard fails to provide consistent QoS during a handover process. This thesis analyses the current standard, and surveys the existing solutions for resolving the issue. Moreover, we propose a new mechanism which is named DAD-Less MIPv6. The mechanism uses a unique but relatively simple method to shorten the handover delay. It is described in detail, and presented with simulation results for demonstrating its performance.

Acknowledgement

During the period of my Masters study, I have received help, guidance and support from many people. It is almost impossible express all my feelings in such a short page.

Firstly, I would like to thank my supervisor Professor Krzysztof Pawlikowski. Thank you for giving the opportunity for me to do this research under your guidance. You have always been positive and kind to me. Your academic knowledge and ability of attention to details have made me a better researcher and thinker. The lessons I have learnt from you would definitely be beneficial for the rest of my life.

Secondly, I would like to thank Professor Harsha Sirisena and Dr Allan McInnes. I really enjoyed taking your suggestions in our weekly meetings. I have kept learning and gaining new perspective in research from both of you. Thank you for keeping challenging me to have more comprehensive view of my research.

Thirdly, I would like to thank Muhammad Arfeen, Lee Begg, Adam Chang, Geoffrey Clark, Mofassir Haque, William Liu, Sayan Ray, Ehsan tabatabaei-Yazdi, Andreas Willig and Huan Zhang . I may not know some of you for very long, but it is indeed a true pleasure to meeting you all during my Masters study. I also would like to thank Gillian Clinton, Peter Glassenbury, Phil Holland, Joffre Horlor and Alex Tobeck for their kind support during my study period.

At last, I would like to thank my girlfriend Vick Li. Thank you very much for your patience and love during my Masters study. Meeting you is one of the best things that have happened during my study period.

Table of Contents

Chapter 1 Introduction

Both academic and industry world foresee the exhaustion of available IP addresses for Internet Protocol version 4 (IPv4) since two decays ago. Many different organizations have discrepancies over exactly when it will occur. In 1994, Internet Engineering Task Force (IETF) forecasted the four billion IPv4 addresses would be fully occupied by 2008. The forecast accorded to the IP usage growth rate at the time. Eleven years later, Cisco Systems reported the available IPv4 addresses would be consumed in 4 to 5 years [15]. In April 2009, an updated report from the Internet Assigned Number Authority (IANA) projected that pool of unallocated addresses would be exhausted in June 2011, with the various Regional Internet Registries using up their allocations from IANA in Oct 2012, [15]. Despite of the differences among the predicted times, the fact of IPv4 exhaustion is unarguable, and the global deployment of Internet Protocol version 6 (IPv6) is imminent.

The shortage of IPv4 address is accelerated due to several factors, one of which is the increasing popularity of wireless technologies. For example, 3G is a family of technology standards which provide speech and data service at high data rate. In 2009, 3G penetration has reached 86.1% for NTT DoCoMo in Japan, [34] and [58].In Europe, there are 910.8 million mobile phone users, and 101.5 million of them are using 3G services. Moreover, China had just launched 3G services in 2009. Currently, most of these 3G services are being carried over IPv4. However, as the number of users grows, the IP address shortage will become more severe. Mobile IPv6 (MIPv6) is the perfect candidate to provide mobility and sufficient addresses that may resolve the issues.

In the last two decades, Internet has evolved from sending electronic messages to transmitting voice and video data. The main traffic stream has become more and more real-time sensible. Therefore, achieving a good quality of service (QoS) has become the main focus of many Internet Service Providers (ISPs). To adopt MIPv6 as the preferable protocol, it is essential that MIPv6 meets the QoS requirements. Currently, the MIPv6 standard fails to provide good QoS in some circumstances. It is especially the case when a mobile device is moving from one MIPv6 network to another, which is called the "handover" process. The network movement creates a delay, which is relatively long for many real-time sensitive applications, such as Voice-over-IP and online gaming. This undesirable behaviour makes the protocol less ideal for many Wireless Internet Service Providers (WISPs). To overcome this shortcoming of the MIPv6 handover process is an important issue for future wireless Internet.

The exhaustion of IPv4 address, the large potential usage of wireless Internet, and the s deficiency of the current handover mechanism of MIPv6 is the motivation of this Masters study. This thesis surveys the existing solutions which aim at shortening the duration of the handover process in the standard MIPv6; and through the study of these solutions and the current standard, a new solution is proposed and tested by simulations.

This thesis contains five chapters. Chapter 1 introduces the research topic and explains the importance of this research topic. Chapter 2 describes the related background knowledge such as IPv6, MIPv6 and types of handover, as well as existing solutions. Chapter 3 presents a new solution which is named DAD-Less MIPv6 with detailed explanation of the mechanism. This chapter discusses the benefits and poten-

tial impacts of the mechanism to the current Internet infrastructure. Later of the Chapter 3, we discuss the benefit of combining the existing solution with the DAD-Less MIPv6 handover mechanism. In Chapter 4, we focus on explaining the assumption of the simulations, and the numerical results are presented in the later part of the chapter. At last, chapter 5 summarises the findings of this Masters study, and propose possible future study which are related to this topic.

Chapter 2 MIPv6 and Its Handover Proposals

When considering the next generation of Internet Protocol, one often refers to IPv6. This chapter provides the background knowledge regarding IPv6, Mobile IPv6, as well as the processes which occur when such a handover is initiated. Later of this chapter also introduces the existing solutions for shortening the duration of the MIPv6 handover.

2.1 IPv6

Internet Protocol version 6 (IPv6) is the next generation of Internet Protocol (IP) which was released by Internet Engineering Task Force (IETF) in 1996. The motivation of the protocol is to resolve the problem of IPv4 address shortage in global Internet. However, the adoption of IPv6 has been slowed by the introduction of network address translation (NAT). The NAT alleviates the address exhaustion by separating the local IPv4 address and the global IPv4 address, and reusing the global addresses locally. However, NAT also makes it difficult and sometimes impossible to use peer-to-peer applications, such as Voice over Internet Protocol (VoIP) and multi-user games. Recently, due to the increasing demand and requirement for the wireless Internet, the deployment of IPv6 has become an urgent issue for the future Internet. Already in August 2005, the U.S. Government has demanded that all federal agencies must deploy IPv6 by July 2010, [52].

In essence, IPv6 offers everything IPv4 does and better, with additional features that were not available with IPv4. The following section lists the specific strong point of IPv6 over IPv4, [4], [57].

1. IPv6 increases the IP address size from 32 bits to 128 bits which can support 10^{28} times more devices in the global Internet. For this reason, it can also allow more levels of addressing hierarchy.
2. Instead of using broadcast, the usage of multicast and "anycast address" in IPv6 provides better scalability of multicast routing.
3. IPv6 has a simpler header format than IPv4 which reduces the processing cost and bandwidth cost.
4. The design of IPv6 is more flexible than IPv4. The header design in IPv6 supports future extensions and new options.
5. The Flow Labelling Capability (FLC) in IPv6 enables the labelling of packets which can be used to optimize QoS. This includes enabling premium pricing for guaranteed delivery, and prioritization of defence or other critical government Internet-based communications, even when network is congested.
6. Unlike IPv4, IPv6 has been designed together with security features. It has Authentication and Privacy Capabilities Extensions to support authentication and data integrity. Also, the IPSec is mandatory to the protocol.

2.2 Mobile IPv6

The Mobile IPv6 (MIPv6) is a standard proposed by the IETF. The official name of standard is "Mobility Support in IPv6", and the last update of the standard is in 2004. As the successor of Mobile IP support in IPv4 (Mobile IPv4), MIPv6 is designed with more experience. It does not only shares many features with Mobile IPv4, but also offers many other improvements. The following list summarizes the major differences between Mobile IPv4 and Mobile IPv6, [17]:

1. In MIPv6, the entity “foreign agent” is excluded which makes the implementation of routers become easier. There is no special support required from the access router any more.
2. MIPv6 has built-in route optimization which belongs to a nonstandard set of extensions in MIPv4.
3. Mobile IPv6 route optimization can operate securely even without pre-arranged security associations. It is expected that route optimization can be deployed on a global scale between all mobile nodes and correspondent nodes.
4. MIPv6 provides support on allowing the route optimization and “ingress filtering” to coexist efficiently on a router. The “ingress filtering” is a technique used to confirm that incoming packets are from the networks they claim to be from. The technique is to prevent denial of service attacks which employ IP source address spoofing. [8]
5. The Neighbour Unreachability Detection which belongs to the IPv6 standard assures the reach-ability from the mobile node to its default router and vice-versa.
6. In Mobile IPv6, when a mobile node is away from its home network, most of packets are sent to it by using an IPv6 routing header rather than IP encapsulation. In result, the amount of resulting overhead are reduced comparing to Mobile IPv4.
7. Mobile IPv6 is decoupled from any particular link layer, as it uses IPv6 Neighbour Discovery instead of ARP which improves the robustness of the protocol.
8. Mobile IPv6 is not required to manage the “tunnel soft state” information because of the usage of the IPv6 encapsulation and the routing header. In a com-

puter network, a tunnel is created by following the tunnelling protocol which encapsulates packets at a peer level or below. It is used to transport multiple protocols over a common network as well as provide the capability for encrypted virtual private networks (VPNs). Inside of a tunnel, when one of the routers encounters an error while processing the datagram, it requires the router to return an ICMP error message to the source of the tunnel. Unfortunately, the size of the ICMP packet is greater than the IPv4 header; it is generally not possible for the router to immediately reflect an ICMP message. To resolve this problem, the source of tunnel requires to maintain extra information regard to the tunnel which is called "soft state" information. [48]

2.3 Types of Handovers

In the wireless network aspect, a handover is usually referred to transferring an ongoing call or data session from one subnet to another. The process can also be known as handoff. A handover process usually causes a transmission to be discontinued in a period of time, so the user may experience a long extra delay for the application he or she is using. During the period, a large amount of packets can be lost depending on the speed of the connection, and the QoS will drop dramatically. Currently, there are many different types of handovers which can be categorized by the connection status, the technology used, the network topology or the layer where they occur in the OSI model.

2.3.1 Soft Handover and Hard Handover

When the handover process is categorized according to its connection status, a handover can be soft or hard. The difference between them is based on whether a mobile device maintains a connection with at least one access points during the handover

process. In the handover period, if the mobile device keeps its connection with the old access point until it fully establishes its connection with a new access point, the handover is called a soft handover. In contrast, if the mobile device breaks its connection with the old access point before it is connected with a new access point, we deal with a hard handover.

2.3.2 Inter-technology Handover and Intra-technology Handover

In the wireless network area, there are many different types of wireless access technology have been developed. Each of the technology provides different connection range, network capability and so on attributes. In future, it is very likely to see all these technologies coexist and complement to each other. Therefore, it may be frequent to see a mobile device using different access technologies while moving. If a handover happens between two different technologies, we deal with an inter-technology handover. Otherwise, it is an Intra-technology handover.

2.3.3 Horizontal Handover and Vertical Handover

Horizontal handover and vertical handover are distinguished by whether a mobile node has changed its access network or access router. The following figure illustrates a horizontal handover.

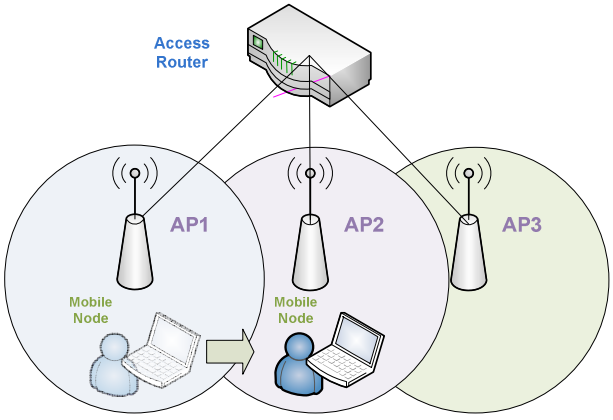

Figure 2.1 Horizontal Handover

Figure 2.1 depicts a mobile node moving from the access range of AP1 to AP2. As the figure demonstrated, both AP1 and AP2 are connected with the same access router. This means there is no topological change from the perspective of the mobile node. Therefore, it is a horizontal handover.

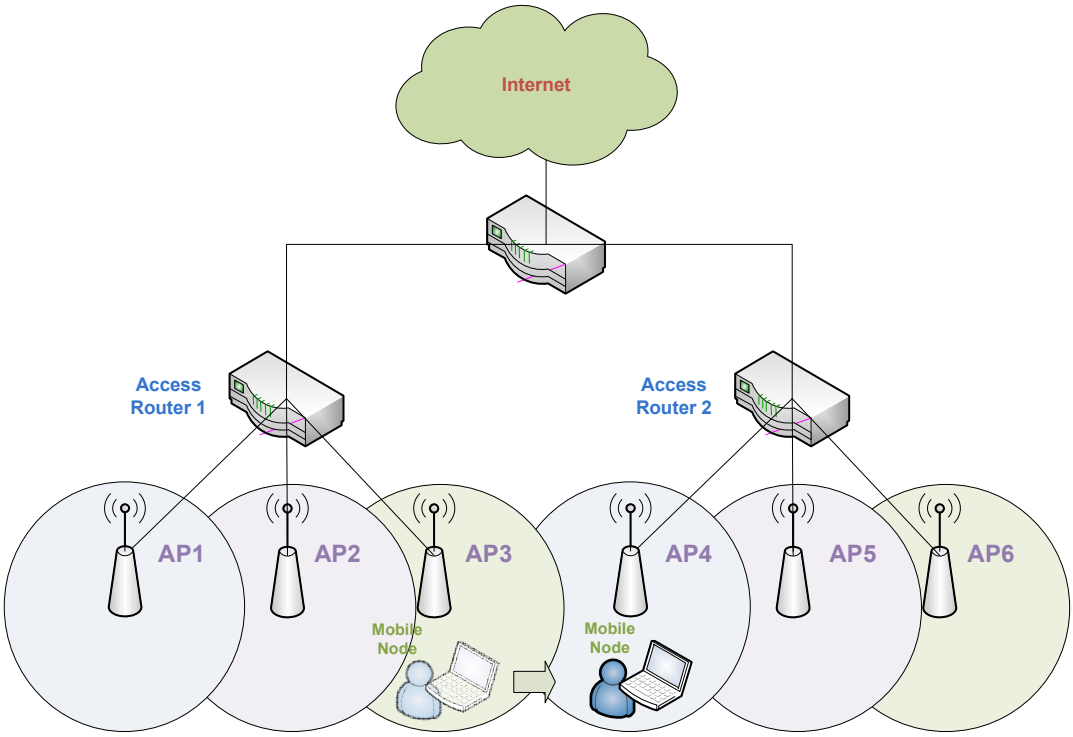

Figure 1.2 Vertical Handover 1

Figure 2.2 demonstrates a scenario of a vertical handover. In this figure, the mobile node moves from the access range of AP3 to AP4. As the figure shown, AP3 and AP4 are connected with different access router. Since the access router of the mobile node has changed, the access network topology is also changed. Therefore, it is a vertical handover.

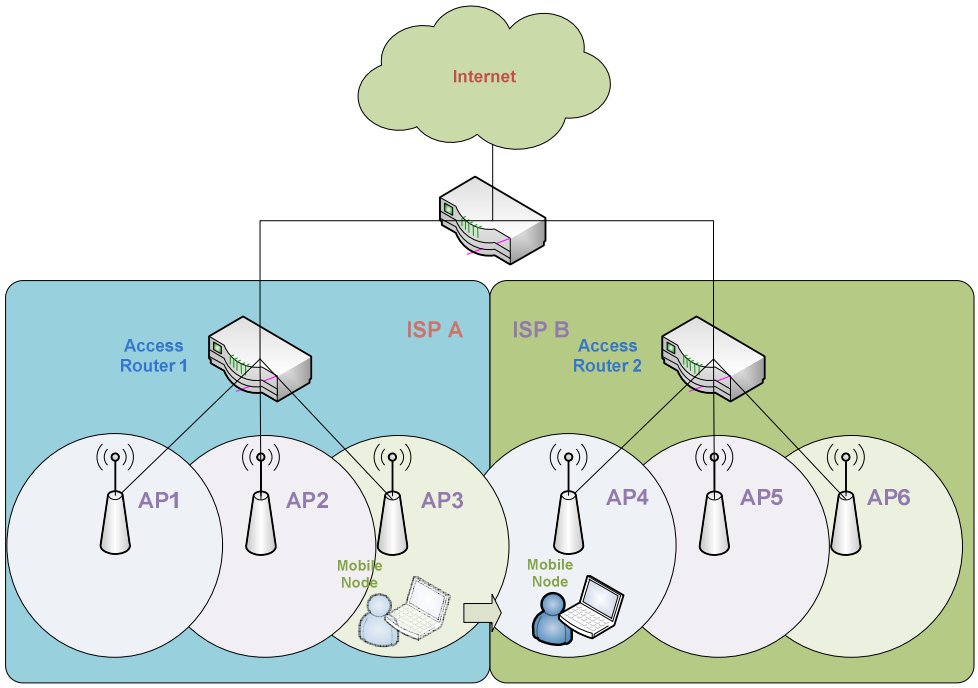

Figure 2.3 Vertical Handover 2

Vertical handover does not only happen within the network of one Inter Service Provider (ISP), it can also happen between ISPs. Figure 2.3 demonstrates such a vertical handover that happens between ISPs. In the figure, the mobile node moves from AP3 to AP4 where AP3 and AP4 are connected with different access routers which belong to different ISPs.

2.3.4 Layer 2 and Layer 3 Handover

A complete vertical handover consists of the processes occurring in layer 2 and layer 3. The processes occurring within layer 2 are known as layer 2 handover, and the processes occurring within layer 3 are called layer 3 handover. The layer 2 handover often indicates the changes of the access point, and the handover delay in this layer is often media or technology dependent. The layer 3 handover often indicates the change of the access route, and the length of the delay is related to the network protocol. A handover in MIPv6 is a layer 3 handover which is the main focus of this research.

2.4 Handover in MIPv6

2.4.1 MIPv6 Terminology and Conditions

This section lists the terminologies which will be used to explain a MIPv6 handover in later sections.

Mobile Node (MN): MN is a terminal that moves between networks.

Access Point (AP): AP is the facility that provides the radio connectivity to MNs.

Access Router (AR): AR is the router that provides Internet connectivity to MNs.

Home Address (HoA): HoA is a unicast address which is permanently assigned to an MN. Usually the traffic will be delivered to the MN by this HoA directly.

Home Agent (HA): HA is the AR that assigns the HoA to an MN. The assigned HoA should have the same network prefix as the HA. The network prefix is a part of IPv6 address.

Home Network (HN): HN is the network where MN has acquired the HoA. It is the network where the HA belongs to.

Care-of-Address (CoA): CoA is a temporary address for an MN while it is not at the HN.

Foreign Access Router (FAR): FAR stands for Foreign Access Router which refers to any AR provides Internet connection to an MN except HA. Please note it is not a Foreign Agent as MIPv4, since there is no special router required in MIPv6.

Foreign Network (FN): FN is the network where the MN is currently connecting with but not HN.

Correspondent Node (CN): CN is the terminal that is currently communicating with the MN.

Figure 2.4 provides a graphical demonstration to all the terms mentioned above.

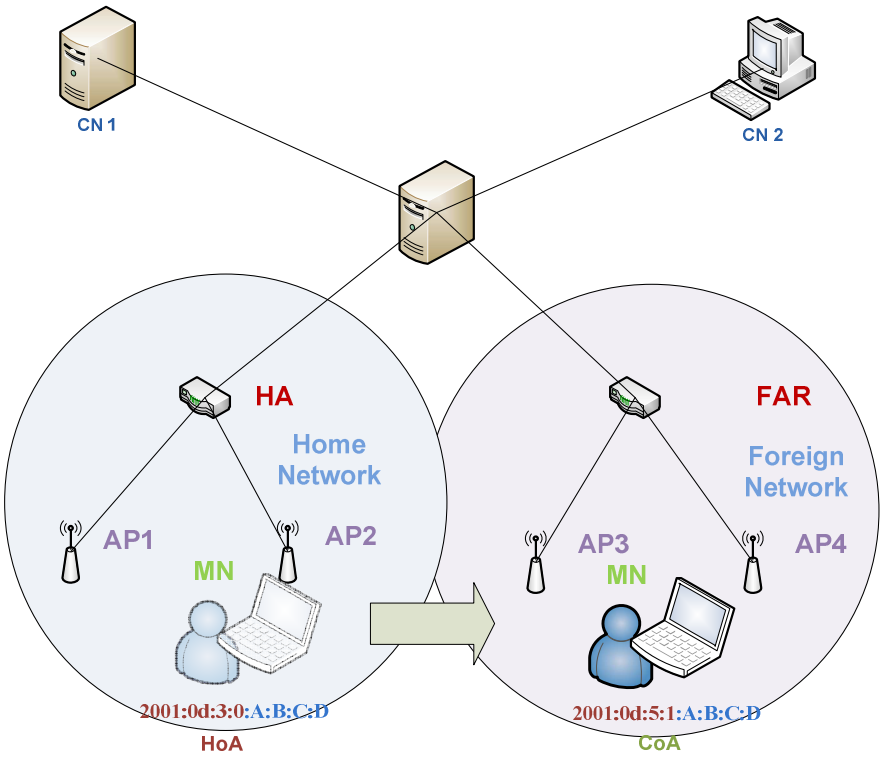

Figure 2.4 Graphical Explanations of the MIPv6 Terminology

In Figure 2.4, the MN (Mobile Node) is labelled with light green colour. It is indicated by an icon which is a combination of a user icon and a laptop icon. In the figure, the MN travels from its HN (Home Network) to a FN (Foreign Network), and range of these two networks are indicated by different colour of circles. Inside of each circle, there are one access router and two access points. In the HN, the access router is the HA (Home Agent) of the MN. In the FN, the access router is referred as FAR (Foreign Access Router). Below the MN's icon, it is the current IPv6 address of the MN. When the MN is at the HN, the MN uses its HoA (Home Address). When the MN is at FAR, the MN uses its CoA (Care-of-Address). The addresses are labelled with different colour in the figure because they represent different parts of an IPv6 address. More details regarding to the address will be presented in the later chapters. On the

top of Figure 2.4, there are two desktop icons which indicate the CNs (Correspondent Node) of the MN. In this context, the CNs are the computers or servers which are currently communicating with the MN. All these terminologies will appear in the explanation of a handover later.

2.4.2 The processes of an MIPv6 handover

A MIPv6 handover can be divided into five different processes: Movement Detection (Movement Detection), Candidate Access Router Selection (CARS), Address Configuration (AC), Authentication & Authorization (A&A), and Binding Update (BU) which are demonstrated in Figure 2.5. Each of these sub-processes is described in detail in the following sections.

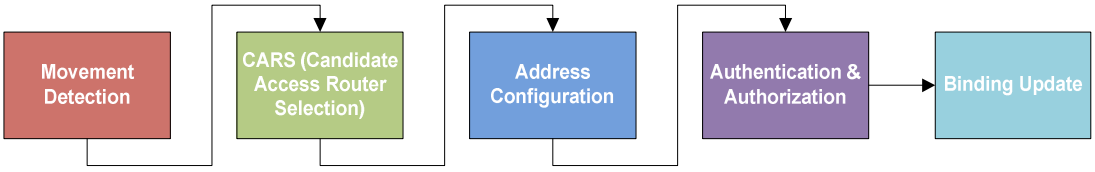

Figure 2.5 Basic Procedures of an MIPv6 Handover

Movement Detection

Movement Detection is a process that recognizes when a Mobile Node (MN) has moved away from its current access network. It is the first stage of a handover. When the movement of the MN is confirmed, a sequence of other handover sub-processes shown in Figure 2.5 will be performed.

According to RFC3775 “Mobility Support in IPv6” [17], the movement of an MN is confirmed when the following two conditions are both satisfied:

1. A new AR has been detected by the MN.

2. The current AR has become bi-directional unreachable. It means that the MN is not able to reach the AR, and the AR is not able to detect the MN either.

These conditions guarantee a handover occurs only when it is necessary. In another words, it means the MN will not perform a handover unless it realizes that the current Internet connection is not available any more. This is one of the reasons why the QoS will drop dramatically during a handover. The Movement Detection process is defined this way to avoid packet loss and signalling overhead during the Binding Update which is the last stage of a handover.

The Movement Detection conditions are tested by the facilities of the IPv6 Neighbour Discovery (ND) which includes the Router Discovery (RD) and the Neighbour Unreachability Detection (NUD).

The Movement Detection process employs two messages from the IPv6 RD messages to confirm the first condition of a network movement. The employed messages are the Router Solicitation (RS) and the Router Advertisement (RA) messages. The mechanism of detecting a new AR behaves as follow. In the MIPv6 wireless networks, every MIPv6 enabled wireless router multicasts a RA message through its APs periodically. The duration of the period is defined by two configurable values MinRtrAdvInterval and MaxRtrAdvInterval in the router. If an MN has been waiting for a RA message from the current AR more than MaxRtrAdvInterval time, the MN will consider it as a movement hint. Then the MN will immediately multicasts a RS message. If any router receives the message, it will reply to the MN with a RA message. This message contains the global IPv6 address of the router and link address of the APs. Once the MN

receives a RA which contains a new IP address of an AR (Access Router), the first condition of the Movement Detection is considered to be satisfied. If the MN receives a RA from a new AR without sending RS message, the first condition is also considered to be satisfied.

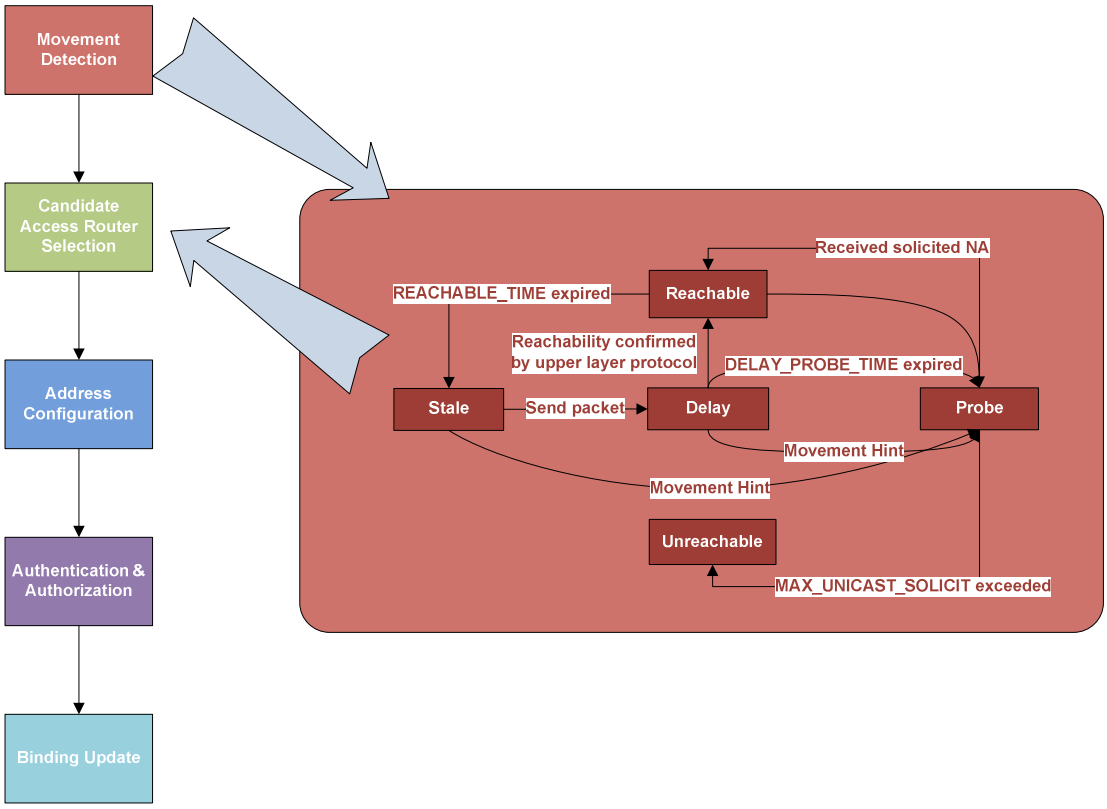

Figure 2.6 State transitions during the execution of the Neighbour Unreachability Detection Procedure

The Neighbour Unreachability Detection (NUD) in IPv6 is used to check for the second condition of a network movement. It verifies the current AR of the MN has become bi-directional unreachable. The behaviour of the NUD are specified by RFC 2461 [33], and depicted in Figure 2.6. According to RFC 2461, every IPv6 node can have five statuses: “Reachable”, “Stale”, “Delay”, “Probe” and “Unreachable”. When an MN enters to a new network, it multicasts ND messages to find the possible neighbours. Once the MN receives a replied message from a neighbour, the state of

the neighbour will be recorded as “Reachable”. After a fix time interval which is known as “REACHABLE_TIME”, the state of the neighbour will change to “Stale”. There will be no further state change until the MN sends a packet to the neighbour. Once a new packet is sent by the MN, the state of the neighbour will be labelled as “Delay”. During the “Delay” cycle, the MN waits for reply from the neighbour for another time interval called “DELAY_FIRST_PROBE”. If the MN does not receive replies from the neighbour within the time limit, the state of the neighbour will change to “Probe”. In this stage, the MN waits a time interval which may take as long as multiple times of the interval between the periodic Neighbour Solicitation (NS) messages. If the MN still does not receive any reply from the neighbour, the state of the neighbour will be changed to “Unreachable”. The waiting interval is exactly specified by MAX_UNICAST_SOLICIT variable times the time interval between the periodic NS messages. The time interval between the periodic NS messages is specified by RETRANS_TIMER variable which can be customized as well as MAX_UNICAST_SOLICIT variable.

In conclusion, the duration of the Movement Detection process essentially depends on the value of MaxRtrAdvInterval, MAX_UNICAST_SOLICIT and RETRANS_TIMER.

The Movement Detection mechanism used by MIPv6 is known as Lazy Cell Switching (LCS). It is a generic solution rather than an optimal solution. There are multiple other movement detection algorithms such as: Early Cell Switching (EyCS), Eager Cell Switching (ErCS) and Enhanced Lazy Cell Switching (ELCS). Since they are not

used by the current MIPv6 standard, the details of the mechanisms are not interest of this thesis. For further details, refer to [38].

Candidate Access Router Selection (CARS)

After a layer 3 movement has been detected by an MN, the MN needs to connect to a new AR to maintain its network connection. The process of selecting a CAR consists of two parts: the Candidate Access Router Discovery process and the Target Access Router Selection (TARS) process, [37], [27].

Candidate Access Router Discovery

CARD is an IETF experimental protocol which has been published one year after the standardization of MIPv6. The specification of CARD is incomplete and it has only been discussed in few papers. The major functions of CARD are acquiring the IP addresses of the Candidate Access Routers (CARs), and discovering the ARs' capabilities.

According to [37], there are two approaches to obtain the CARs' IP addresses. The first approach relies on the MN to listen to the Layer2 (L2) beacon messages from a new Access Point (AP). The AP's L2 ID contained in the beacon messages can map to the IP address of the CAR. The mapping process requires the MN to maintain a connection with the CAR. In general, one network interface can only allow the MN to connect with one CAR. This means the MN is required to have multiple network interfaces to maintain connections with different CARs. This is not desirable for cost efficiency or hardware design, [6].

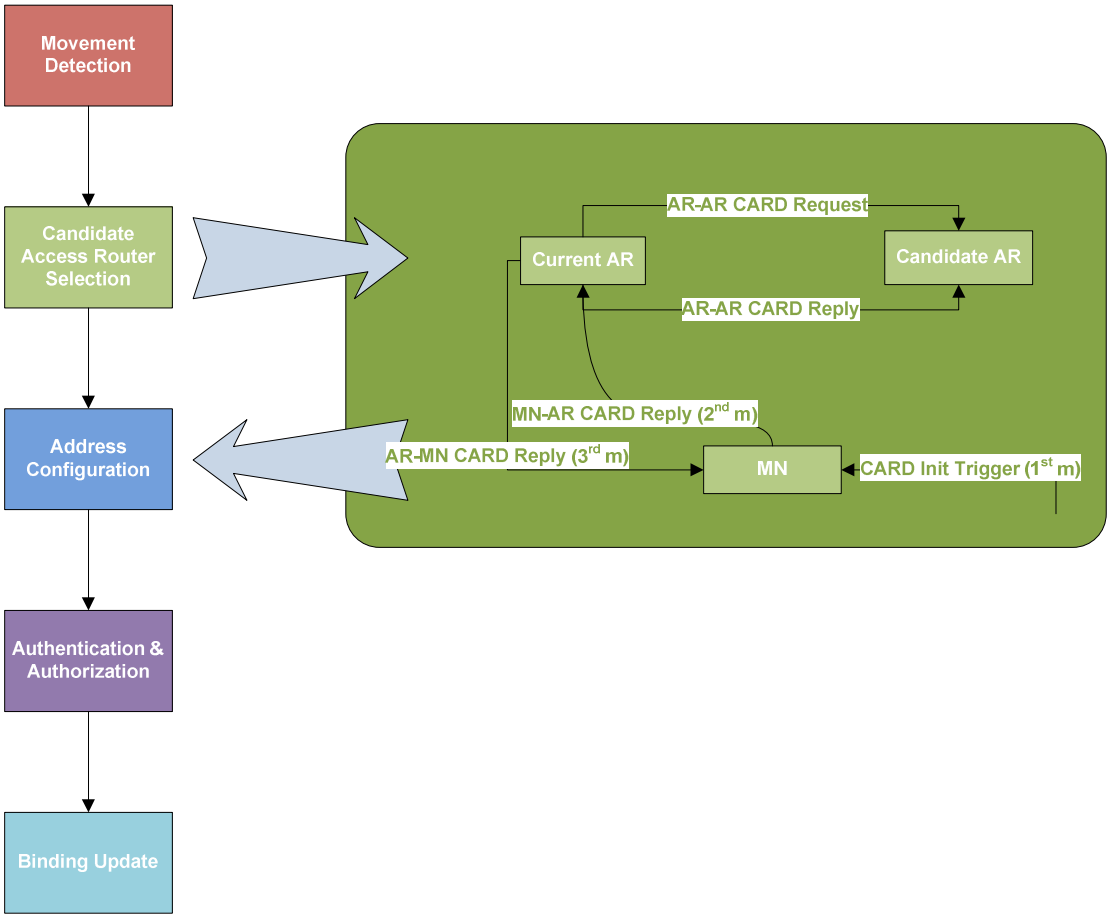

Figure 2.7 Processes of the MN-initiated CARD Protocol

The second approach is the standard solution which requires the assistance from the current AR. Once an MN receives a beacon message from a new AP, the CARD process will be initiated. The MN sends an MN-AR CARD request which passed the Layer 2 ID of the new AP to the current AR. If necessary, the MN may set a C-flag to request the new CAR's capability in addition to its IP address. The current AR then performs reverse address translation to resolve the address and the capability information of the CAR if required. The current AR obtains the information by the AR-AR CARD Request/Reply messages. Through the messages, every AR updates other ARs' table entries of the Layer 2 ID and their capability periodically. Therefore, when an MN requests the information, it will be available from AR's local CAR table, [20].

Target Access Router Selection (TARS)

TARS can be performed by either the MN or the current AR. The capability information of the CARs which are obtained from the CARD process is fed into the TARS process. The TARS process uses specific algorithm to choose the most appropriate Access Router (AR). The capacity information includes information about such properties of the CARs as: bandwidth, available channels and so on. Since there is no standard algorithm for this process, the process will not be explored any further in this thesis. For more details, refer to [6].

Address Configuration

After the new AR has been selected, the MN will need a new temporary IPv6 address according the RFC 3775. The process of acquiring of the address is called Address Configuration (AC). The temporary address is usually known as Care-of-Address (CoA).

According to [17], [50], there are two approaches to obtain a CoA for an MN. One is the stateless address configuration, and the other is the stateful address configuration. The stateful address configuration is usually performed by DHCPv6 [54] which behaves in a similar way as DHCPv4. The method in general appears to be too time consuming for a handover. Therefore, the stateless address configuration is usually preferred.

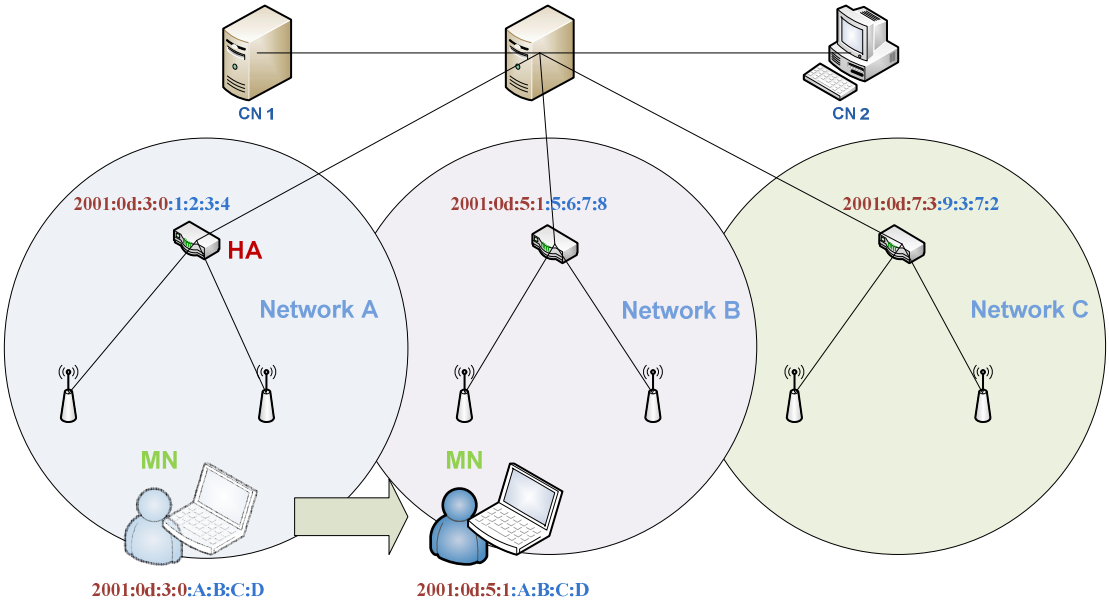

Figure 2.8 Example of an Address Configuration Process

In a wireless IPv6 network, every MN and sub network has an interface identifier. The stateless address configuration forms the CoA by combining the prefix of the network and the prefix of the MN. This is demonstrated in Figure 2.8. In this figure, the red part of IP addresses is the network prefix, and the blue part is the MN prefix. After the MN has moved from network A to network B, the IP address of MN has changed its network prefix only, but keeps using the MN's prefix which is the second part of the CoA.

A CoA can be used only after it has passed the Duplicate Address Detection (DAD), and this process has been standardized by IETF, [51]. The standard only defines how to detect whether a CoA is unique, but it does not mention the procedures after a duplicated address is found. "A tentative address that is determined to be a duplicate as described above, MUST NOT be assigned to an interface and the node SHOULD log a system management error. If the address is a link-local address formed from an interface identifier, the interface SHOULD be disabled", [51]. Disabling an MN, when

it fails the DAD is not a desirable solution in practice. For most of the Wireless Internet Service Providers (WISPs), it is more logical to use stateful address configuration as a backup procedure.

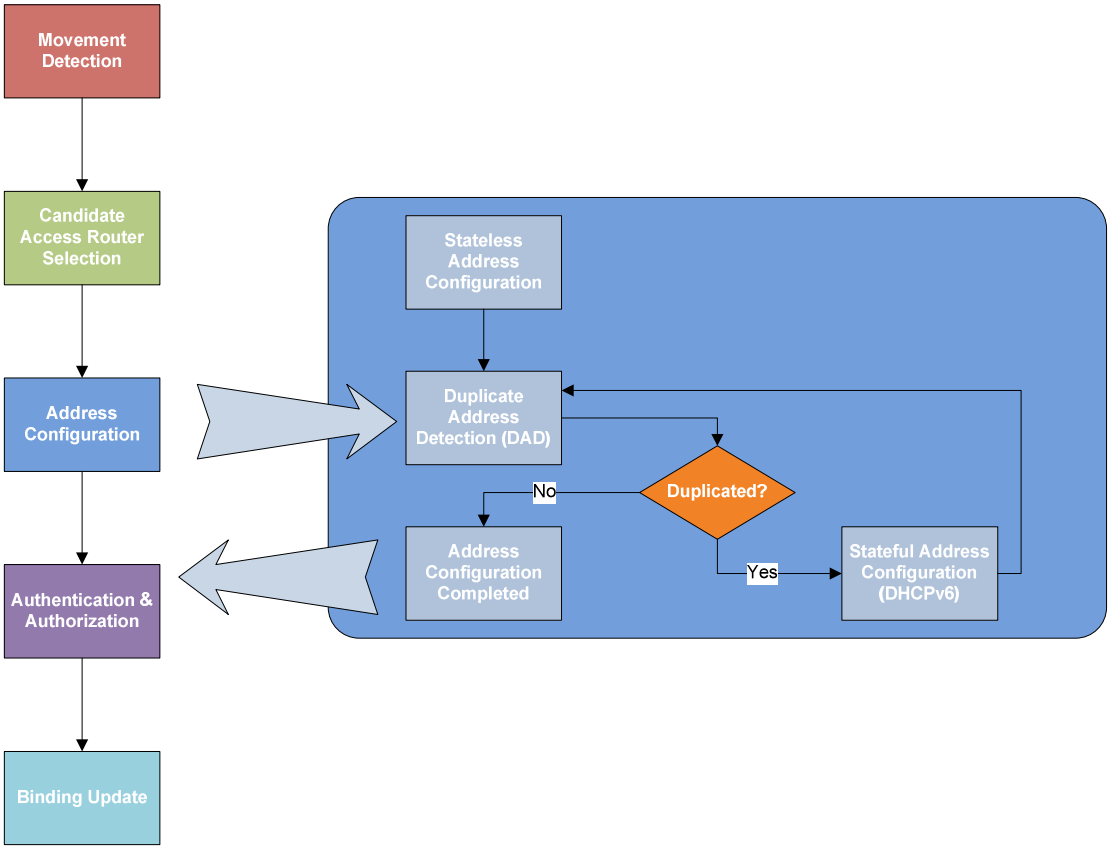

Figure 2.9 Sub-processes of the Address Configuration

Figure 2.9 demonstrates all the states in an address configuration process. The process starts with a stateless address configuration which is used to form a CoA. The newly formed CoA will be tested by the DAD, and if the address is duplicated, a stateful address configuration will be performed. The stateful address configuration will assign a new CoA to the MN, and address will be tested by the DAD again. This cycle repeats until the assigned address passes the DAD. Once the address has been confirmed to be unique, the handover will shift to another stage which is Authentication and Authorization.

Authentication and Authorization (A&A)

The A&A process is used for checking whether an MN has the authority to use the connection from an AR. Since the security of MIPv6 is not the main focus of this thesis, the process will not be discussed in detail.

Binding Update

Bind Update (BU) is the last stage in a handover process. The purpose of the BU is to keep tacking the network location of an MN for its Home Agent (HA) and the Correspondent Nodes (CN). The BU process is completed with assist of two messages which are a BU message and a Binding Acknowledge (BACK) message.

a. Binding Update message to HA

Inside of every HA, there is a Binding Cache Entries (BCE) table where records both the HoA and CoA of MNs. The BCE allows the MNs to be reachable in Internet, and it is frequently updated by BU messages.

b. Binding Update to CN

The BU is sent to CNs only when the "route optimization" mode is used in IPv6. According to the MIPv6 standard [18], there are two communication modes between MN and CN. One is "bi-directional tunnel" mode, and the other one is "route optimization" mode.

In the "bi-directional tunnel" mode, MNs are not required to register on its CNs. HA is the only node that keep tracking the location of MNs. When a CN intends to send a

packet to an MN, the packet will have to be delivered to the HA of the MN first. The HA then will redirect the packet to the MN. Conversely, if the MN tries to send a packet back to the CN, the packet will need to be sent to the HA first. The HA then will redirect the packet to the CN.

In the "route optimization" mode, the HA is excluded from the packet delivery. The CN keeps a BCE itself for tracking the location of the MNs. In this case, any packets between CNs and MNs are transmitted directly. The "router optimization" mode obviously saves more network resource and reducing the round trip time between the MN and the CNs. Therefore, in general, for a MIPv6 handover, the "router optimization" mode is used, and the BU messages are sent to both HA and CNs.

c. Binding Acknowledgement (BACK)

If a BU message has been successfully received by an MN's HA or a CN, the HA or the CN will reply a BACK message to the MN. When an error occurs in the process, the HA or the CN will send a Binding Error message. For further detail, please refer to [17].

Till here, we have finished describing all the stages in a MIPv6 handover. Then we will be able to understand how the existing solutions can shorten the duration of a MIPv6 handover.

2.5 Survey of existing handover protocols

A MIPv6 handover can cause delay and/or packet-loss for an ongoing traffic stream. Over last ten years, a number of research projects have aimed at shortening the handover process by improving different part of a MIPv6 handover. There are improvements to Movement Detection process were considered in [9], and [37] for the CARS

process, [44], [41], [19] and [26] for the AC process, [44], [49], [20], [41], [56] and [5] for the BU process. The following section presents the proposals which mostly have been standardized by IETF. They are mainly improving the AC process and the BU process.

2.5.1 Fast Handover MIPv6 (FMIPv6)

FMIPv6 is a protocol which has been standardized by IETF to improve the AC process in a MIPv6 handover, [23]. There are two operation modes: (i) the predictive mode, (ii) the reactive mode.

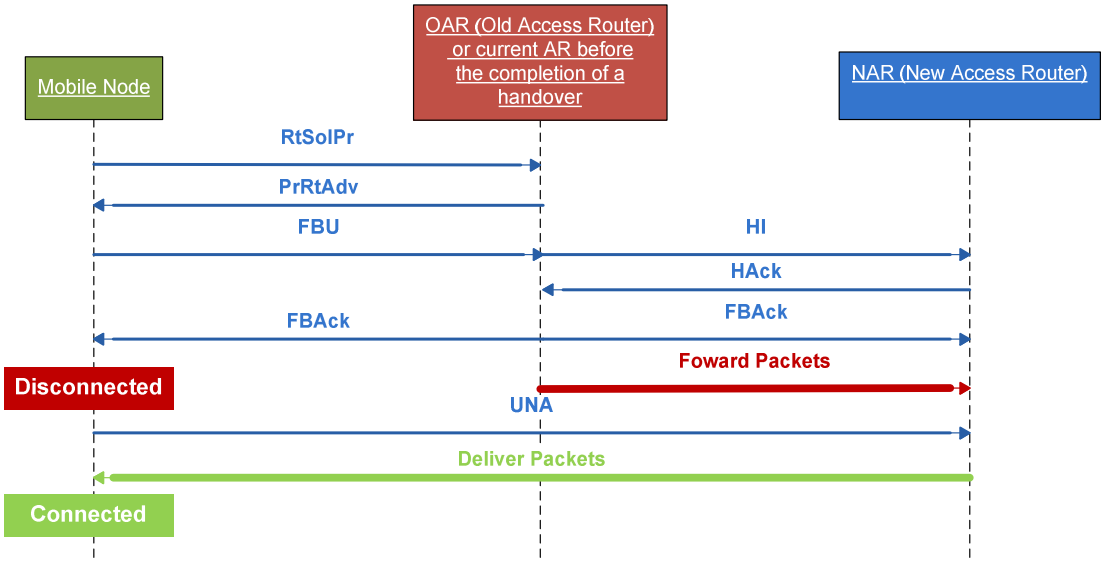

Figure 2.10 Predictive Mode of FMIPv6

The predictive mode

The predictive mode shortens the AC process by performing the process before it is required. This means the handover process is started without the confirmation from a standard Movement Detection process. In addition, the packet loss is minimized by buffering packets at the Old AR (OAR) and forwarding them to the New AR (NAR).

Figure 2.10 illustrates the steps in this mode, and details and functionality of each step is described in the following paragraphs. For more information, refer to [23].

a. **Router Solicitation Proxy (RtSolPr) and Proxy Router Advertisement (PrRtAdv):** Router Solicitation Proxy (RtSolPr) and Proxy Router Advertisement (PrRtAdv) messages are transferred between an MN and its current AR in the beginning of an FMIPv6 handover. The RtSolPr message is sent from the MN to its current AR, and the message is used for requesting the information from a NAR. As Figure 2.10 shown, the current AR will be also referred as OAR (Old Access Router) after the MN has connected to the new AR. After receiving the message, the current AR will reply to the MN with a Proxy Router Advertisement (PrRtAdv) message which contains the AP-ID (Access Point ID) of the found AP, the IPv6 address and network prefix of the NAR (New Access Router) which the AP belongs to. After receiving the PrRtAdv message, the MN is able to form a new CoA that will be used in the new access network.

b. **Fast Binding Update (FBU):** The Fast Binding Update (FBU) message is sent from the MN to the OAR after receiving the PrRtAdv message. The FBU message contains the new CoA which is derived from the information that is contained inside of the PrRtAdv message. The OAR extracts the new CoA from the FBU message, and binds the new CoA with the current CoA for forwarding packets to the NAR.

c. **Handover Initiation (HI), Handover Acknowledge (HAck) and Fast Binding Acknowledge (FBack):** After the OAR receives the FBU message, the OAR sends the Handover Initiation (HI) message to the NAR. The HI message contains the current CoA, the layer 2 address and the proposed CoA of the MN. The NAR uses the information to execute the AC process which contains a DAD process,

and may be a stateful address configuration process if the proposed CoA is duplicated in the network. This AC process is exactly same the standard MIPv6 AC process which has been described in previous sections. Then, the NAR replies to the OAR by a Handover Acknowledge (HAck) message. The message is used to inform the OAR that the AC process has been completed successfully. During the AC process, if the proposed CoA fails the DAD, the HAck message may also suggest a CoA. The suggested CoA is derived from the stateful address configuration process. The CoA will be forwarded by the Fast Binding Acknowledge (FBack) message from the OAR to the MN.

d. **Unsolicited Neighbour Advertisement (UNA):** The UNA message is the last message in an FMIPv6 handover which is shown in Figure 2.10. The MN breaks its connection with the OAR after receiving the FBack message. The OAR then will forward the buffered packets to the new CoA of the MN. The packets will be buffered at the NAR until a UNA message is sent from the MN. Once the UNA message is received by the NAR, the NAR will remove and update internal entries, and the buffered packets will be forwarded to the MN.

The reactive mode

If the overlap area between two APs from two different AR is relatively small, and the MN moves too fast, the MN may be able to receive the FBack message before the OAR becomes unreachable. In this case, the predictive mode will have never enough time to be performed complete, and the reactive mode is specially designed for this situation.

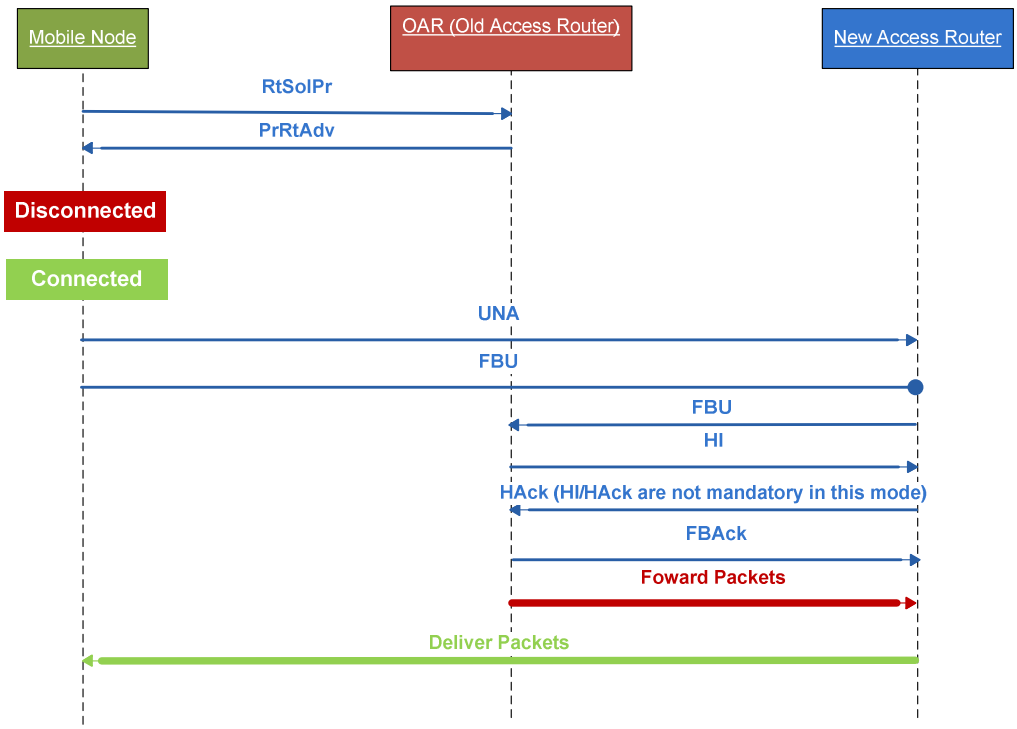

Figure 2.11 Active Mode of FMIPv6

The unexpected disconnection of the OAR causes the FMIPv6 to reorder its sequence of messages. The biggest change is the order of the UNA message. The UNA message is no longer the last message sent in a fast handover, but the first message after the MN connects to the NAR which is shown in Figure 2.11. Following the UNA message, a FBU message is also sent to the NAR, and forwarded to the OAR. From this point, the message sequence becomes the same as in the predictive mode. According to the FMIPv6 standard, the HI/HAck message pair may not be required for quickening the process. At last, the OAR will send an FBack message to the NAR with the buffered packets, and the packets will be forwarded to the MN.

The main difference of the predictive mode and the reactive mode are when the UNA message has been sent in an FMIPv6 handover and where the FBU message is sent. The reactive mode almost does not shorten the handover latency comparing to the

standard MIPv6, but minimizes the packet loss by using a tunnel to forward the packets from the OAR to the NAR.

In conclusion, the main difference between FMIPv6 handover and MIPv6 handover is when the AC process is initiated. The standard MIPv6 requires two conditions to confirm a layer 3 movement which leads to the initiation of the AC process. However, FMIPv6 starts the AC process once an AP from a new access router has been detected.

2.5.2 HMIPv6

Hierarchical Mobile IPv6 (HMIPv6) is another standard for improving the MIPv6 handover process which has been published in early 2000s and updated by IETF in 2008, [49]. It is an extension of MIPv6 like FMIPv6, and it focuses on reducing the latency caused by the BU process in a MIPv6 handover.

After an MN changes its AR, the MN needs to update its new CoA to its HA and CNs to maintain its reach-ability. The BU messages are specially designed for this purpose. However, when the MN is geographically too far from the HA or the CNs, the BU process may add hundreds of milliseconds to the handover process because of the propagation delay. HMIPv6 is specifically designed to solve this problem.

Instead of sending the BU messages to the relatively far HA and CNs, in HMIPv6, the MNs send the BU messages to a closer proxy server. The mechanism is demonstrated in Figure 2.12 below.

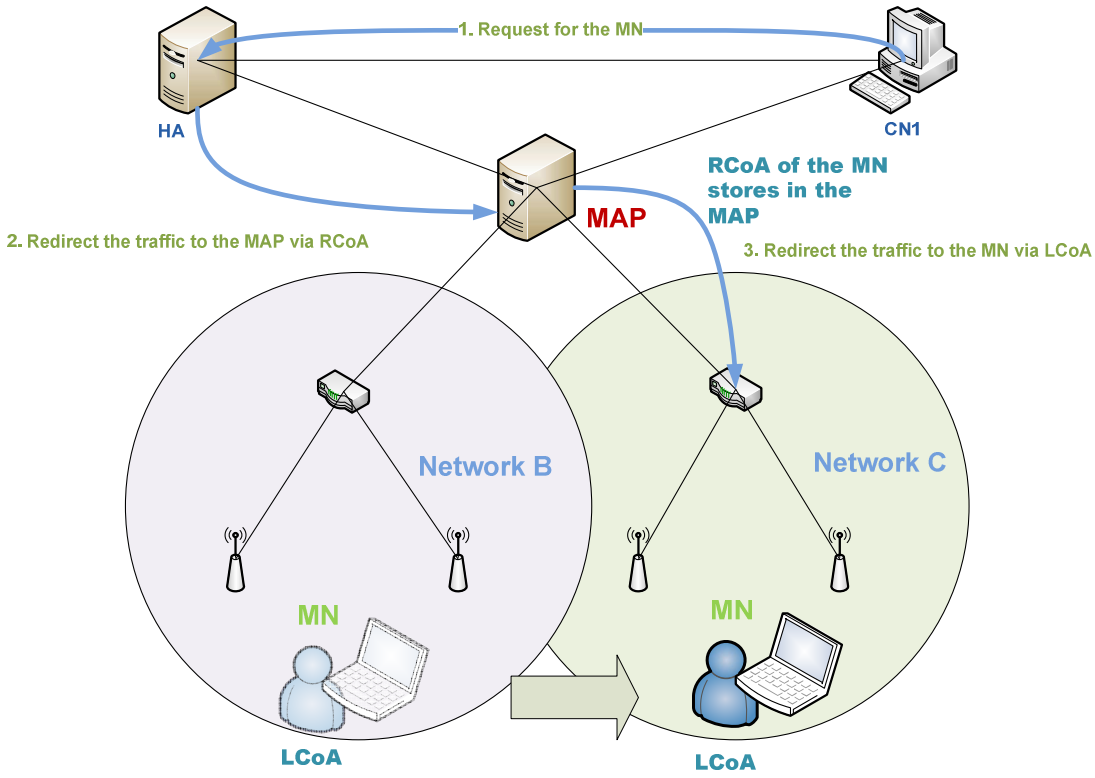

Figure 2.12 Example of HMIPv6 in action

In Figure 2.12, there are three new elements added to the standard MIPv6. They are MAP, Regional Care-of-Address (RCoA) and Local Care-of-Address (LCoA). MAP is the proxy server which stands for Mobility Anchor Point (MAP). It behaves like a HA which keeps tracking the location of an MN within its domain. The RCoA and the LCoA are used to allow an MAP to work with an MN's HA and the CNs seamlessly. When an MN enters to a MAP's domain, an RCoA and an LCoA are both assigned to the MN. The RCoA is sent to the MN's HA instead of the normal CoA within a BU messages. From this point, as long as the MN moves within the domain of the MAP, the MN only needs to change its LCoA. The BU process only happens between the MN and the MAP, and the LCoA is stored inside of the MAP for updating the MN's location. The HA will not these BU processes until the MN leaves the domain, and changes its RCoA.

In MIPv6, from the perspective of a CN, every time when a CN intends to communicate with an MN, the node will send a packet to the MN's HoA, and the traffic will be directed to the MN's HA. If the MN is not within the HN, the HA will search the binding entries to match the HoA and the CoA. In the HMIPv6, the CoA is substitute with an RCoA. Via the RCoA, the traffic will be directed to the MAP by the HA. After the traffic is received by the MAP, the MAP will search its binding entries to match the RCoA and the LCoA. At last, the traffic is sent to the MN according to the LCoA.

2.5.3 FHMIPv6

FHMIPv6 is another proposed standard which was published by IETF in 2006, [25]. The proposal is essentially a combination of FMIPv6 and HMIPv6 which has not been standardized till the beginning of 2010.

In order to allow the FMIPv6 and the HMIPv6 to work together, the original messages involved in a handover need to be modified to fit this purpose. Figure 2.13 below illustrates a handover process involved in FHMIPv6 in the predictive mode.

Same as FMIPv6, the MN exchanges information with the OAR by the Router Solicitation Proxy (RtSolPr) and the Proxy Router Advertisement (PrRtAdv) messages. Similar to FMIPv6, once the MN receives the hint for a handover, the Fast Binding Update (FBU) message is sent to the OAR. In addition, the FBU message is also sent from the OAR to the MAP. Comparing to FMIPv6, it is the MAP that communicates with the NAR by the HI/HAck pair messages instead of the OAR. At the same time, the OAR will buffer all the incoming packets for the MN while the MN is detached from the OAR. The MAP sends a Flush message to the OAR to indicate that the HI/HAck messages have been sent successfully between the MAP and the NAR, and

the BCE will be updated in the MAP. Once the BCE is updated, the MAP will send the FBack message to the NAR with a new LCoA for the MN, and it will also forward all the packets to the new LCoA. Meanwhile, the OAR sends all the buffered packets to the NAR and end with a Flush message. Once the MN is connected to the NAR, the MN will send a FNA message. Then the NAR will forward all buffered packets and future incoming packets to the MN.

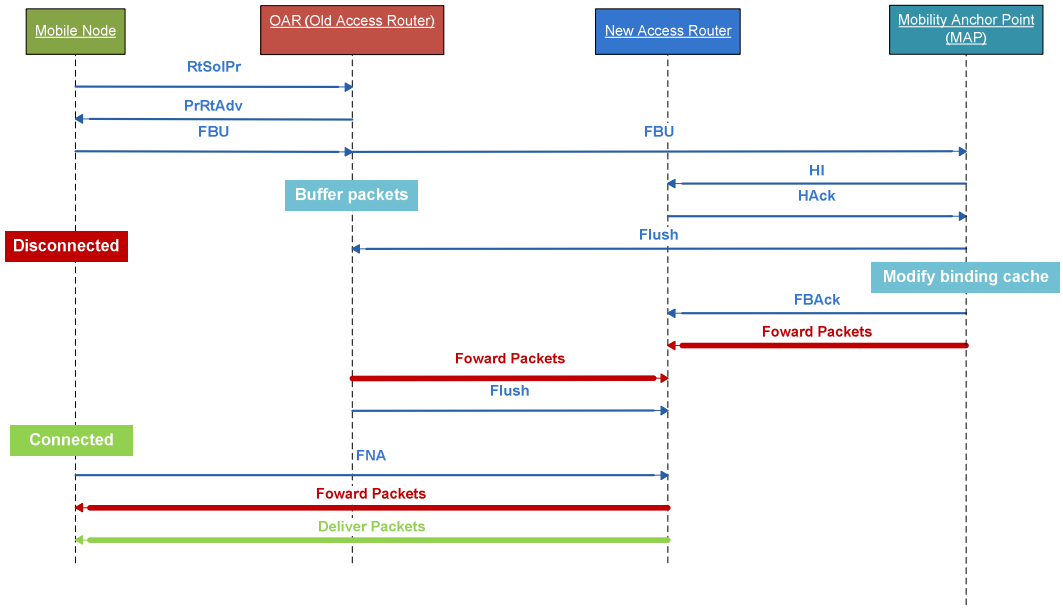

Figure 2.13 Predictive Mode of FHMIPv6

The reactive mode in the FHMIPv6 is activated when an MN has not been able to send a FBU message to the OAR to continue the predictive mode. This is mode is very similar to the FMIPv6 reactive mode. Once the MN attaches with the NAR, the MN will send a FNA message to the NAR with its information. It is the NAR's responsibility to inform the OAR that a fast handover is required. The NAR send a FBU message to the OAR, and the OAR will start to buffer the incoming packets for the MN. Since this point, the reactive mode has same message sequence as the predictive mode from the point that the OAR sends the FBU to the MAP.

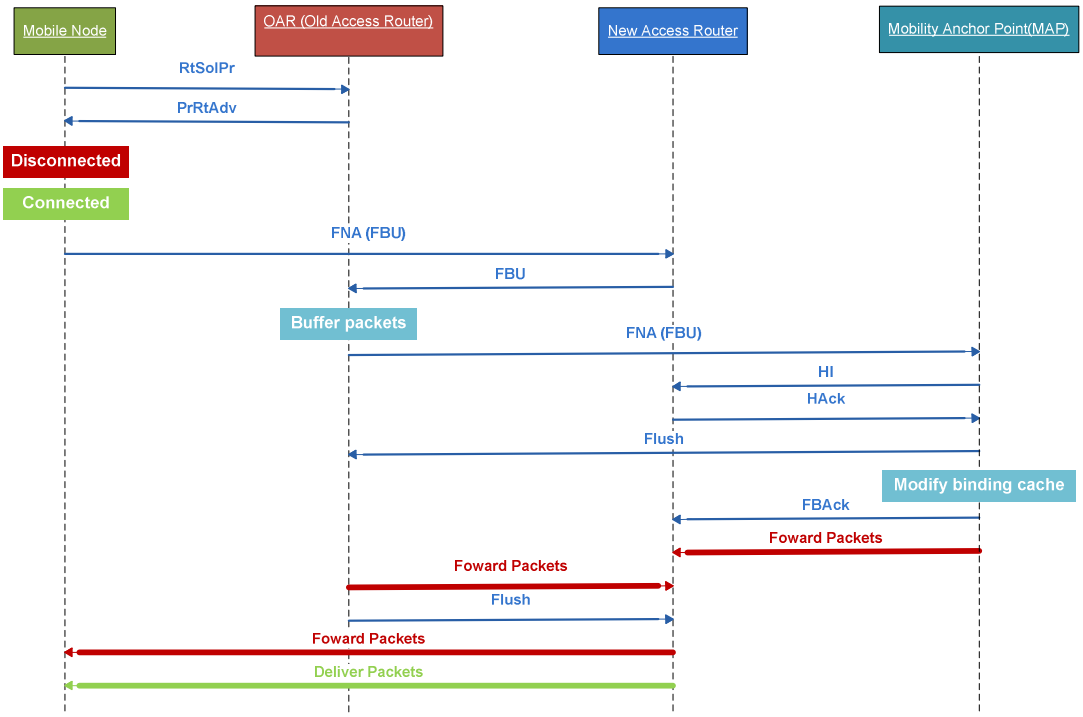

Figure 2.14 Reactive Mode of FHMIPv6

2.5.4 Seamless MIP (S-MIP)

Seamless MIP (S-MIP) was originally designed for the MIPv4 standard [44], and has been proposed to work with IPv6, [45]. S-MIP adopts the idea of combining the HMIP and the FMIP, and provides further improvements in two aspects. Firstly, S-MIP includes the consideration of the Candidate Access Router Selection (CARS) process which is neglected by other proposals. (CARS is described in the first chapter.) This is achieved by using three routers to track the physical location of the MNs, and a server to determine the best Candidate Access Router respectively. Secondly, the mechanism intends to avoid the case which the forwarded packets may be out of order during a handover. This is achieved by separating the forwarded packets from the different sources, and sending them at different chunk of time. The details of the proposal are described in the following paragraphs.

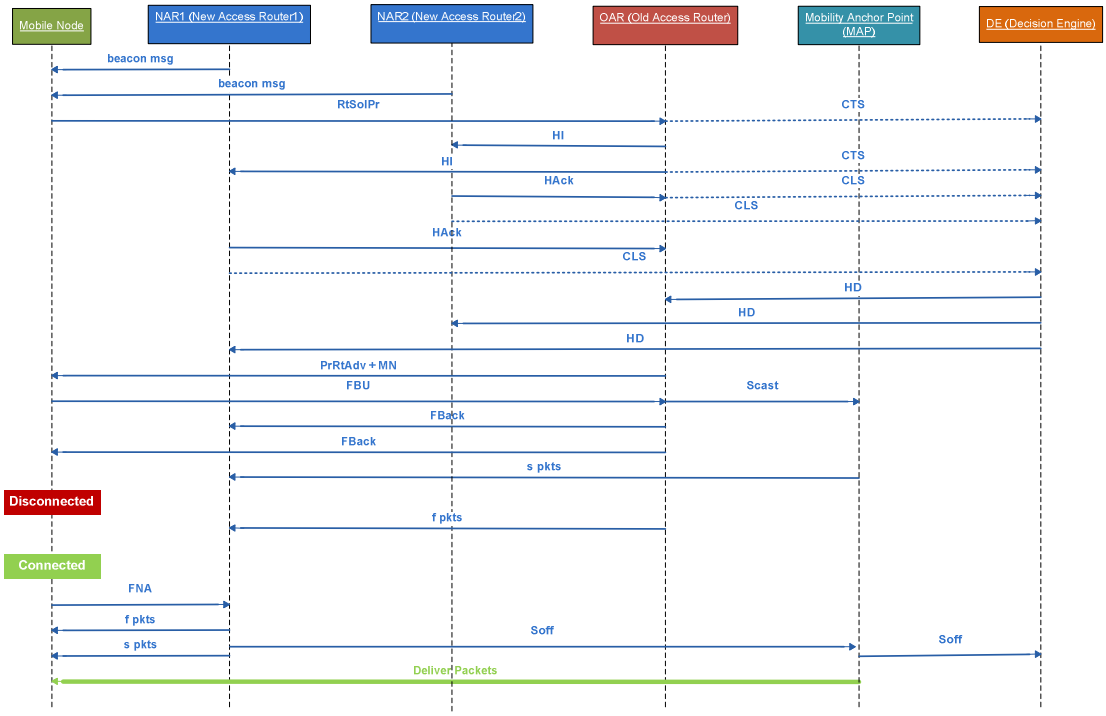

Figure 2.15 The message sequence of S-MIP

As Figure 2.15 shown, the S-MIP has one extra entity and multiple new messages compare to FHMIPv6. To make the concept more clearly, the names and the functionalities of the entity and the messages are listed below.

Current Tracking Status (CTS) message is used to track the physical location of the MN for DE.

Carrying Load Status (CLS) message contains the information regarding to the loading capacity of the AR.

Handoff Decision (HD) message contains the handover decision information from the DE. It informs the entire candidate ARs which AR has been chosen for the MN.

Handoff Notification (HN) message is sent from the OAR to the MN. It indicates which NAR the MN should connect to. The content of the message is derived from the HD message.

Simulcast (Scast) message is the message which starts the Synchronized-Packet-Simulcast (SPS) process. The SPS process is used for separating the packets from the OAR and the MAP. By doing this, there will be less possibility of massing the order of the packets.

Simulcast Off (Soff) message terminates the SPS process.

Decision Engine (DE) is the entity which uses the information from the CTS and CLS messages to make the handover decision.

A handover process in the S-MIP starts with detecting the NARs from an MN. Once the MN receives beacon messages from the NARs, the MN will send an RtSolPr message to the OAR. The OAR will immediately sends the CTS message to the DE to update the location of the MN. Then the OAR sends HI message to all the NARs, and the NARs will send their own CTS message to the DE too. All the CTS messages will be used for the DE to work out the exactly physical location of the MN. In response to the HI message, all NARs will send back a HAck message back to the OAR, and the CLS messages are also sent to the DE for the handover decision. After the DE makes the decision basing on the capacity loading status of each NAR and the location of the MN, the DE sends the HD messages to the OAR and all the NARs for the result of the decision. HN and PrRtAdv messages will be returned to the MN by the OAR. The MN replies a FBU message to the OAR, and the OAR will send a Scast message to the MAP to start the SPS process. The OAR will also reply a FAck message to the selected NAR and the MN each. At the mean time, because of the SPS process, the buffered packets from the NAR and the MAP are labelled with “f” and “s” header respectively. Once the MN completes the L3 handover with the NAR, the NAR will forward the “f” packets and “s” packets in sequence. Each forwarding process is ter-

minated with the Soff message. At last, the MAP will be able to forward all incoming packets directly to the MN with the new CoA.

2.5.5 PMIPv6

PMIPv6 is a protocol originally proposed by CISCO in 2007, [10]. It is a network-based mobility management protocol rather than a host-based protocol like MIPv6. This means the MNs are not involved with any process of the mobility management. The protocol intends to optimize both the AC and the BU processes. The AC process is improved by emulating a home network environment to the MNs all the time. The MN is unaware of any subnet changes since the NAR behaves exactly like MN's HA. Therefore, the MN can keep using its HoA, and the DAD process is no longer required. The BU delay is shortened by using a similar approach as HMIPv6. The BU messages are sent to a close server instead of the MNs' original HAs. The detail of this protocol is explained in the following paragraphs.

In order to achieve the improvements mentioned above, PMIPv6 introduces two new entities to the original MIPv6. They are the Local Management Anchor (LMA) and Mobile Access Gateway (MAG). The LMA behaves like a MAP which tracks the network location of the MNs. In addition, the LMA also provides information to the MAG for emulating the home link for every MN which are inside of the LMA's domain.

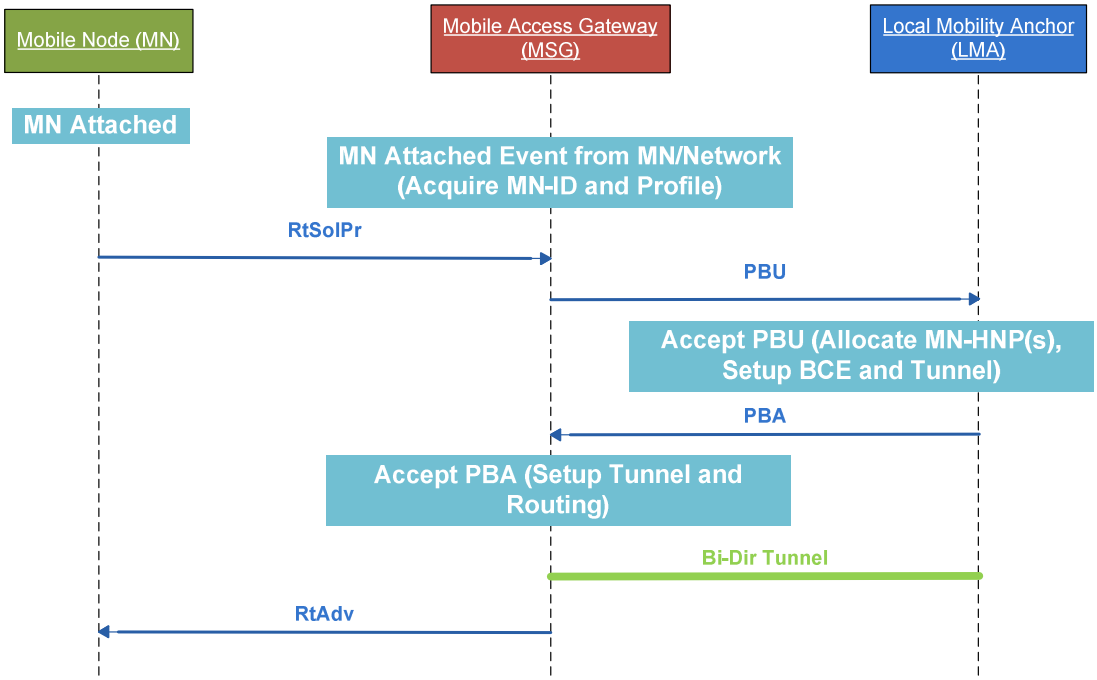

Figure 2.16 Processes occurring during MN Registration in a PMIPv6 Domain

Figure 2.16 illustrates the events and messages occurred during an MN registration to a PMIPv6 network. When an MN first time enters a PMIPv6 network, the MN will be notified an event "MN Attached". Meanwhile, the MAG detects the attached MN, and start to acquire MN-ID and profile information. Then the MN requests the MAG's information by sending a Router Solicitation (RtSol) message. Different from MIPv6, the MAG sends a Proxy Binding Update (PBU) to the LMA first instead of replying the Router Advertisement (RtAdv) message to the MN immediately. The LMA uses the information contained inside of the PBU message to update its Bind Cache Entries (BCE) for recording the location of the MN, and request for setting up a bi-directional tunnel. The request of the tunnel and the Home Network prefix(es) of the MN are sent back to the MAG by the Proxy Binding Acknowledge message. Once the PBA is accepted by the MAG, the bi-directional tunnel is set, and the MAG replies the MN with

the Router Advertisement (RtAdv) message. The RtAdv contains the MN's Home Network prefix(es), and the MN will consider it is at the Home Network.

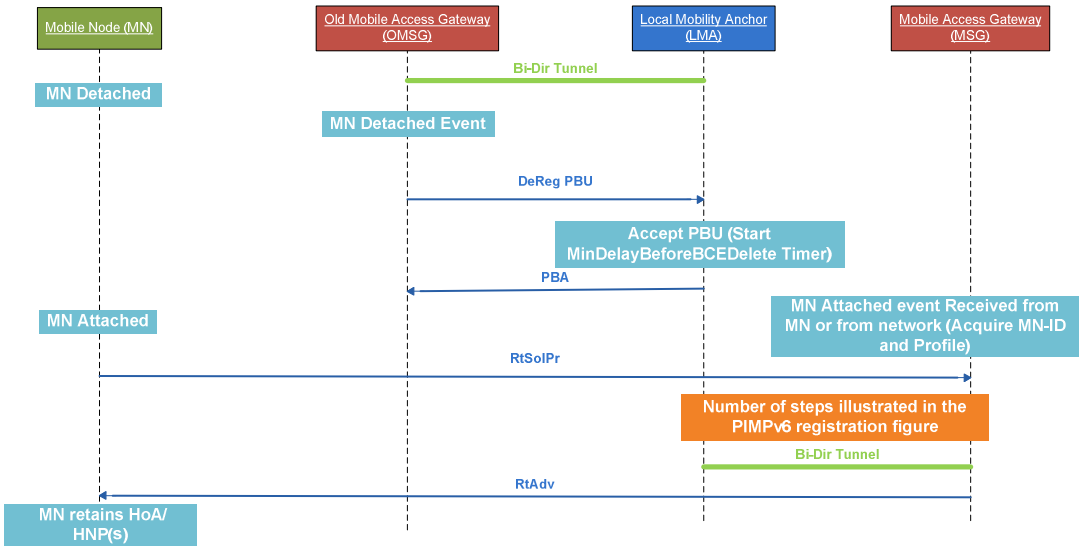

Figure 2.17 Processes of a handover in a PMIPv6 domain

When the MN changes the access network within a LMA domain, the handover procedure follows the above figure. Once the MN has been detached from the Old Mobile Access Gateway (OMAG), both the MN and the OMAG will be notified. The OMAG then sends a De-Registration Proxy Binding Update (DeReg PBU) message to the LMA for updating the location of the MN and removing the bi-directional tunnel. The LMA waits for the "MinDelayBeforeBCEDelete" amount of time for the New Mobile Access Gateway (NMAG) to updates the location of the MN. If the timer expires, the MN's location entry will be removed from the BCE. Then the LMA replies the OMAG with the PBA message. If a NMAG does reply to the LMA before the timer expires, the registration process has been described in the last paragraph will be repeated. However, it is not always true like the figure shown that the MN's attachment to the NMAG is after its detachment with the OMAG.

2.5.6 Comparisons of the existing solutions

There are multiple publications which compare the performance of the existing proposed solutions. However, most of these comparisons are evaluated before the newest MIPv6 standard was proposed in 2004. The simulation models used in the publications are no longer accurate. Especially the previous simulation models have never been compared with the test-beds at the time when they were developed, so the accuracy of the models according to the older standard also remains uncertain. Furthermore, each of the suggested solutions has been proposed for a specific application. Therefore, the performance of these solutions is scenario-dependent which makes them hard to compare. For example, the HMIPv6 shortens the handover process by improving the Binding Update process, and the FMIPv6 shortens the handover by performing the Address Configuration process before it is required. Since the duration of the Binding Update process strongly relies on the propagation delay of the network, the HMIPv6 may have a better performance than FMIPv6 in some cases, but worse in others. Precise specification of applications which perform better under specific handover mechanism will be possible when their accurate simulation models become available. Currently, the only credible MIPv6 simulation models are those in OMNeT++ [35] and probably in OPNET [36].

2.6 Summary

This chapter introduces the background knowledge of IPv6 and types of handover in the early part. After that, we discuss MIPv6 handover in detail. Each stage of a MIPv6 handover has been discussed and explained. Later, the chapter surveys the existing solutions which include five mechanisms proposed for improving the MIPv6 hand-

over process. Each mechanism is briefly explained with their working concepts and necessary procedures.

Chapter 3 DAD-Less MIPv6

3.1 Source Delays in handovers of MIPv6

As mentioned in the previous chapter, a standard MIPv6 handover can be distinguished with processes on: Movement Detection (MD), the Candidate Access Router Selection (CARS), the Address Configuration (AC), the Authentication & Authorization (A&A) and the Binding Update (BU). Each of these processes extends the duration of the handover process, so improving any one of them can be beneficial for better QoS during this process.

In the standard MIPv6, the movement in the network layer is confirmed when two conditions occur: the Mobile Node finds a new AR and the old AR has become unreachable. Some proposals attempt to shorten the handover by simplifying the MD requirement, for example the FMIPv6 and other its variants. However, when there are many wireless service subscribers in a small area, a single AR will certainly not be able to serve them all. Therefore, it will require many ARs which will have a large amount of overlap to resolve this problem. In this situation, if an MN confirms a network layer movement every time when a new AR is found, the MN will perform many extra handovers while its current connection is still acceptable. This will consume MN's power by sending signals too frequent for performing unnecessary handovers. Therefore, excluding one of the two indications of movement may not be ideal. The simple but effective solution for speeding up the MD process should be increasing the frequency of sending the RtAdv messages; for more details, refers to Chapter 2. However, the frequency of sending these messages should be within a certain limit for saving the MNs' power and ARs' resource. The MIPv6 standard has already sug-

gested such restrictions. Therefore, improving the MD process is not the focus of this research.

The CARS process consists of the Candidate Access Router Discovery (CARD) process and Target Access Router Selection (TARS) process. Adding extra information in the RtAdv message could allow these two sub-processes to be performed in parallel with the MD process. The latency caused by this process can be considered as one of most obvious latencies to be eliminated. However, since a little research on the sub-processes themselves has been done, both processes are not standardized by IETF yet. Improving the handover process by modifying the CARS process remains an option for future investigation.

The AC process contains the stateless address configuration, the DAD, and if necessary, the stateful address configuration too. In the AC process, the delay is mainly caused by DAD process. This process is used to eliminate the possibility that the assigned address may be duplicated in the access network. To avoid these delays, the FMIPv6 intends to perform the AC process before it is required. The proposal eliminates the delay in the predictive mode, but there is not improvement regarding to the latency in the reactive mode. Finding the solution to allow the reactive mode to be faster or even simplify the FMIPv6 can be beneficial to the handover process.

The A&A process falls into the network security area which is out of the scope of this thesis. Therefore, it has not considered either in this thesis.

The last process occurring in a MIPv6 handover is the BU process. The duration of a handover resulted from this process is mainly the propagation delay. In the HMIPv6, there is a new entity which is a proxy server (MAP). It is usually closer to the MNs comparing to the HA. By using MAP and two temporary addresses, the propagation delay is minimized. The mechanism is simple and also accurately for resolving the issue. An optimized handover mechanism may use this idea from the HMIPv6.

In summary, the MD process can be improved by increasing the frequency of sending the RtAdv messages within certain limit as suggested in MIPv6. The CARS can be shorten or even eliminated by being performed in parallel with the MD process. The A&A process is not the main scope of this study, and the BU process has been optimized by the HMIPv6. The following sections describe a new approach to improve the AC process which leads to less MIPv6 handover latency.

3.2 The DAD-less Mobile IPv6 handover mechanism

MIPv6 has been proposed after the MIPv4. When MIPv6 was designed, it had heavily borrowed ideas from its previous version, and the DAD process is one of them. The length of this process time is random generated between 0 to 1 second by the uniform distribution. Quite often the generated delay is relatively long for a MIPv6 handover. To be able to skip or remove the DAD process from a MIPv6 handover would potentially shorten the process averagely by 0.5 seconds which greatly improve the QoS condition during the handover process. This fact has been noticed already in literature; see [1], [26], [53]. In attempt of reducing the DAD related delay has been proposed [26], by minimising the duration of DAD in optimistic duplicated address (ODAD) procedure. A more drastic and simplistic solution was proposed in [1], which suggested to remove DAD completely from the existing MIPv6 standard. The research

stated the probability of an Interface Identifier collision among the nodes sharing a link can be lower than the probability of network equipment outage. Finally, [53] is a modified version of FMIPv6 which also removes the DAD process from a MIPv6 handover. This method requires implementing extra messages in a MIPv6 handover process and introducing new functions to the access router. The following sections introduce an alternative solution which suggests multiple minor modifications to the standard MIPv6 to eliminate the DAD process delay completely. We also discuss the feasibility of the solution, and compare the solution with the existing solutions. At last, we describe how to conjunct the solution with the existing handover solutions.

In order to remove the DAD process from a MIPv6 standard requires modifications in the initial address assignment process, the AC process for the Care-of-Address, the BU process, the HA and the AR.

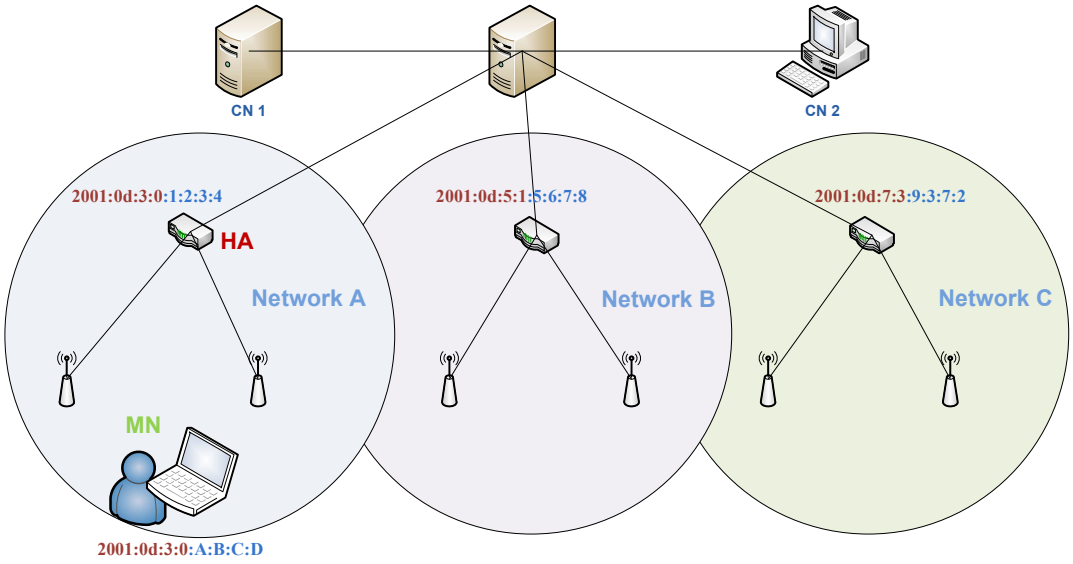

Figure 3.1 Initial address assignment in a MIPv6 based wireless network

3.2.1 Initial Address Assignment and address management

The initial address assignment is not a part of a handover process. However, the initial address assignment is crucial for eliminate the DAD process in a handover.

a. Initial Address Assignment in Standard MIPv6

When a user connects to a MIPv6 based wireless network, his or her machine (MN) will be assigned with an IPv6 address. For example, in Figure 3.1 above, the user is connected to the Network A with an assigned IPv6 address **2001:0d:3:0:A:B:C:D**. In the figure, each address is presented by two different colours. The first part of the IPv6 address is in dark red colour. It is the network prefix address. The second part of the IPv6 address is in light blue, and it is known as MN prefix address. The address can be formed by either stateful address configuration or stateless configuration. The address stateless configuration assigns the address by combining the network prefix and a predefined MN's prefiex. The stateful address configuration follows the DHCPv6 protocol. The assigned address is the combination of the network address prefix and a temporary generated prefix address. Usually the address of a MN is assigned by the stateless address configuration.

b. Initial Address Assignment in DAD-Less MIPv6

In the DAD-Less MIPv6, the initial address has to be assigned by the stateful address configuration (DHCPv6), and the assigned address has to be unique on the Internet. Currently, every ISP owns a range of IP addresses which are assigned by the Internet Assigned Numbers Authority (IANA). In the DAD-Less MIPv6, within the assigned addresses from the IANA, the ISP can use the DHCPv6 protocol to distribute addresses to their users automatically and permanently. The assigned address for each user will be occupied in the address pool until the user switches off the MN or unsubscribes the service. In this case, the functionality of an IPv6 address in a DAD-Less MIPv6 network is essentially equivalent to a mobile phone number in a mobile phone network. Therefore, each IPv6 address of a MN is ensured to be unique from the very beginning.

3.2.2 Care-of-Address Configuration

a. Care-of-Address Configuration in the Standard MIPv6

In a standard Mobile IPv6 wireless network, when an MN moves to a new network, the MN will be assigned with a new CoA. Usually, it is assigned by the stateless address configuration process. The new CoA is obtained by combining the prefix of the network and the prefix of the MN. In Figure 3.2 below, the MN IPv6 address has changed from **2001:0d:3:0:A:B:C:D** to **2001:0d:5:1:A:B:C:D**. The network prefix has changed because of the movement of the MN. This CoA will be checked by the DAD process in the standard Mobile IPv6. If the new CoA is already existed in the new network, the router will process the stateful address configuration (DHCPv6). The cycle repeats until a locally unique CoA is obtained.

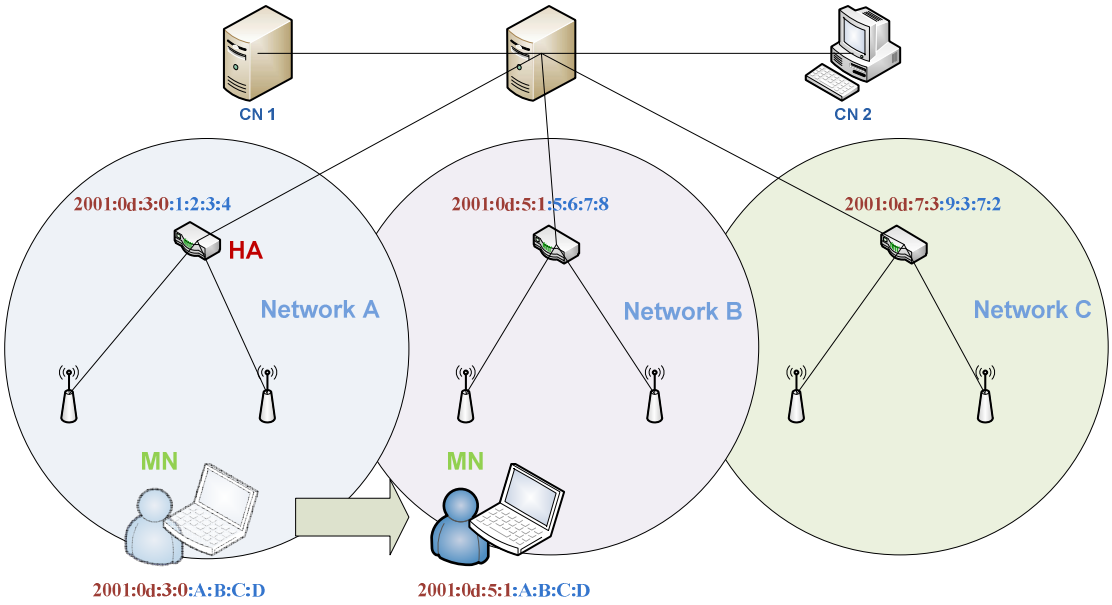

Figure 3.2 A MN moves to another standard MIPv6 wireless network

b. Address Configuration in DAD-Less MIPv6

In a DAD-less Mobile IPv6 wireless network, every Mobile Node is assigned with a unique global IPv6 address. The unique address is used in all access networks, so the

DAD process and the CoA generation are no longer required. As Figure 3.3 demonstrated, the address of the MN has not been changed after MN moved from the Network A to Network B. In DAD-Less MIPv6, the candidate AR only needs to register the IPv6 address of the MN to the local address table, but not required to perform address configuration.

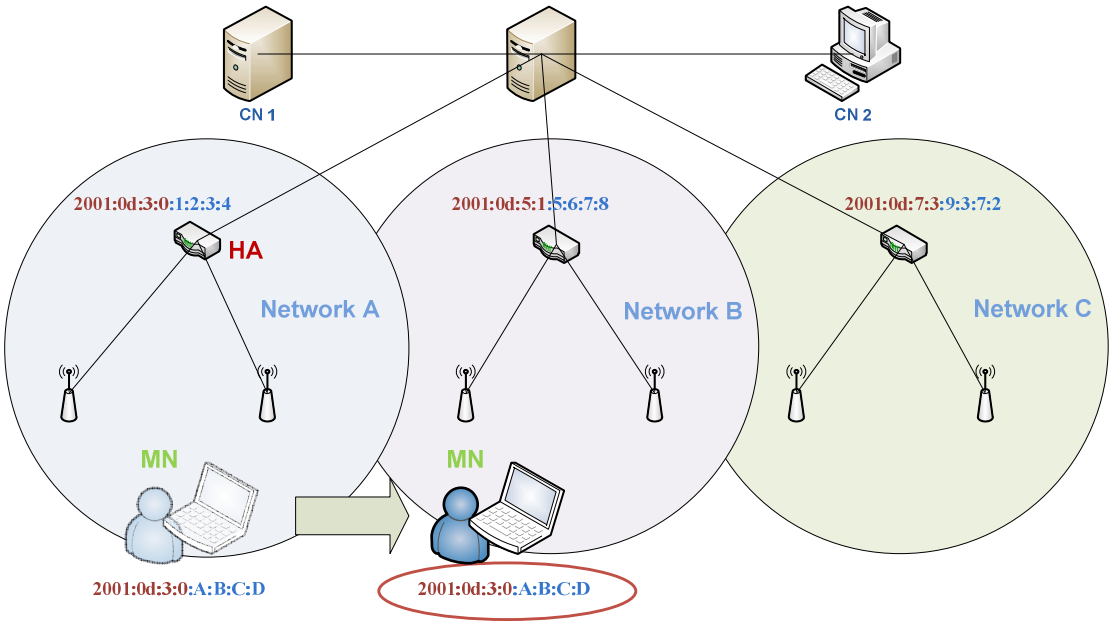

Figure 3.3 When the MN moves to a DAD-Less MIPv6 wireless network

3.2.3 Binding Update

a. Binding Update in Standard MIPv6

In the standard MIPv6, both HA and CNs keeps a list of binding cache entries (BCE) which matches the HoA and CoA of the MNs. The BU process is responsible for updating this BCE. At the end of every handover, the BU messages are sent to the HA and CNs from the MN, and the newest CoA will be updated in the entry. Table 3.1 is an example of a HA's BCE in MIPv6. As Table 3.1 shown, each entry holds the HoA and CoA together, so the HA and CNs can keep tracking the network location of the MN.

Home Address	Care-of-Address	Sequence Number	Lifetime	Flags
2001:0d:3:0:A:B:C:D	2001:0d:8:2:A:B:C:D	20	250	A/H/K/L
2001:0d:3:0:E:F:G:H	2001:0d:5:1:E:F:G:H	57	400	A/H/L

Table 3.1 An example of MIPv6 binding cache entries

b. Binding Update in DAD-Less MIPv6

The only difference in the BU messages used in the case of the DAD-Less MIPv6 is the updated address. The BU messages in the DAD-Less MIPv6 records the address of the current AR instead of the CoA. This is demonstrated in Table 3.2 below. The main functionality of a CoA is to keep tracking the network location of a MN. Therefore, the MN can be accessed even when it is away from the Home Network. By recording the address of the AR of the MN, the network location of the MN can be recorded without using a CoA.

IP address	Access Router Address	Sequence no.	Lifetime	Flags
2001:0d:3:0:A:B:C:D	2001:0d:3:0:1:2:3:4	20	250	A/H/K/L
2001:0d:3:0:E:F:G:H	2001:0d:5:1:5:6:7:8	57	400	A/H/L

Table 3.2 An example of DAD-Less MIPv6 binding cache entries

3.2.4 Modification in a Home Agent (HA)

a. Home Agent in the Standard MIPv6

One of the main responsibility of the HA is to redirect the traffic to the MN when it is away from the Home Network. When a random node in Internet tries to establish a connection with an MN, the node will connect to the HA of the MN first. If the MN is away from the Home Network, the HA will redirect the traffic to the MN via the CoA. This is achieved by inserting the CoA into the routing header of the first requesting packet. Once the connection is established between the MN and the node, the HA will be involved in further data transmission.

b. Home Agent in the DAD-Less MIPv6

In the DAD-Less MIPv6, the function of CoA is achieved by the address of the current AR. Therefore, when a HA intends to redirect a traffic, it inserts the current AR's address to the routing header instead of the CoA.

3.2.5 Modification in an Access Router

a. Access Router in Standard MIPv6

An AR is usually at the edge of a network. It keeps a local address table which maps the local IP addresses with the MAC addresses. For the normal AR, the table usually only keeps the addresses which have the same network prefix as the access router. Table 3.3 is a typical example of such table. All the recorded local address has the same network prefix which is in dark red colour.

Local Address	Mac Address
2001:0d:8:2:A:B:C:D	00-1A-A0-C2-1C-81
2001:0d:8:2:E:F:G:H	02-3C-B0-C2-C3-59
2001:0d:8:2:I:J:K:L	00-4B-05-B3-2A-3C
2001:0d:8:2:I:J:K:L	5B-2A-3C-4D-1C-AA

Table 3.3 An example of local address table in a standard MIPv6 access router

b. Access Router in DAD-Less MIPv6

In the DAD-Less MIPv6, the AR records the unique global address of the MNs, so the network prefix may not be all the same. Table 3.4 is an example of the address table in the DAD-Less MIPv6. In the table, the fourth row has different network prefix from other rows. It is an MN which is away from its Home Network. By recording the global address of an MN in the local address table of an AR, the AR will perform a local delivery if it receives a packet for the MN. Therefore, the data will still be able to reach the MN without using the CoA.

Local Address	Mac Address
2001:0d:8:2:A:B:C:D	00-1A-A0-C2-1C-81
2001:0d:8:2:E:F:G:H	02-3C-B0-C2-C3-59
2001:0d:8:2:I:J:K:L	00-4B-05-B3-2A-3C
2001:0d:3:0:I:J:K:L	5B-2A-3C-4D-1C-AA

Table 3.4 an example of local address table in a DAD-Less MIPv6 access router

3.2.6 Summary of modification of the standard MIPv6 handover

The main principle of DAD-Less MIPv6 is ensuring the IPv6 address of any MN is unique from the very beginning on entire Internet. The address of the current AR for an MN substitutes the role of the CoA in the proposed solution. The BU messages have a minor modification in the content. They update the address of the current AR instead of the CoA of the MN. Due to the changes to the standard MIPv6 protocol, the HA and the AR used in this proposal have required minor modifications too.

3.3 Feasibility Discussion

The previous sections have described the required changes in MIPv6 in order to eliminate the DAD process. This section discusses the feasibility of each change in practice.

The first condition of the DAD-Less MIPv6 is assigning every MN with a unique IPv6 address on Internet. In IPv4, there are nearly four billion available addresses only. Thus, it is obviously insufficient for six billion peoples on the planet. However, the situation is different in IPv6. IPv6 has been originally design to counter IP addresses deprivation. The 2^{128} possible addresses in IPv6 can allow every human on the planet today to have access to 5.6×10^{28} addresses. Furthermore, ISPs in the world are assigned with different ranges of IPv6 address by IANA. With a proper programmed DHCPv6, assigning a unique IPv6 address to each MN on the planet should not be hard to achieve.

The second change to the standard MIPv6 is the exclusion of the AC process for the CoA. The major functions of this process are assigning a new CoA to an MN and per-

forming the DAD on the CoA. Since the CoA is no longer required, the AC process is no long required too. This change makes the implementation of a MIPv6 AR easier than before. While DAD-Less MIPv6 reduces the handover latency, it also simplifies the implementation of the ARs.

The third change is regarding to the BU process. The BU in the DAD-Less MIPv6 behaves almost exactly the same as the standard MIPv6 expect that the address it records. Instead of recording the new CoA, the BU updates the access router address to the BU entries. From programming point of view, it may only require to change one variable. Therefore, it should also be easy to achieve.

The fourth change is that the inserted address in the routing header by the HA. The inserted address is the address of the current AR instead of the CoA of the MN. Again, this modification only requires changing a variable in the related software.

The last change is related with requiring the AR to be able to store foreign IPv6 addresses. This change may even do not require any firmware update.

When a new Internet protocol is designed, one needs to consider its scalability, implementation and maintainability. DAD-Less MIPv6 removes the DAD process and the CoA from MIPv6 protocol which is no major change to the standard MIPv6 itself, so the DAD-Less MIPv6 can be as scalable as MIPv6. Benefitting from the simplification, DAD-Less MIPv6 should be easier to implement than MIPv6. At last, the DAD-Less MIPv6 is a variant of MIPv6, it preserves the maintainability of the MIPv6.

Due to all the reasons above, DAD-Less MIPv6 seems to be a feasible solution for shortening the MIPv6 handover process.

3.4 Comparison between FMIPv6 and PMIPv6

Among the existing proposals mentioned in the chapter 2, FMIPv6 and PMIPv6 have been standardized, and they both improve the AC process in a handover as DAD-Less MIPv6. Therefore, FMIPv6 and PMIPv6 are chosen to compare with the DAD-Less MIPv6 in this section.

3.4.1 FMIPv6 and DAD-Less MIPv6

FMIPv6 shortens the handover latency by performing the AC process before it is required. The approach does eliminate the latency caused by the AC process, but it also introduces new issues. Firstly, the FMIPv6 adds five extra messages to assist the AC process. The extra messages create signal overheads which complicate the process and consume more energy from the MNs. If there are multiple MNs which are trying to perform handovers in the same wireless network, extra messages may cause more interference. Secondly, performing the AC process before it is required can suffer from the Ping-pong movement. (Ping-pong movement describes a movement pattern which moves back and forth in a certain area.) When an MN is having a Ping-pong movement in the overlap area of two or more ARs, the MN will perform many unnecessary address configurations. Consequently, there will be more delays, packet loss and interference. The signal resource and AR resource in the wireless networks will be wasted. The MN may suffer a noticeable amount of power loss too. Thirdly, if an MN moves too fast before a fast handover is completed; the FMIPv6 has to use the reactive mode which causes same amount of delay as the MIPv6, although the packet-loss are less than the MIPv6.

The DAD-Less MIPv6 improves the handover by removing the AC process completely. In the DAD-Less MIPv6, there are no extra messages are required as FMIPv6. Therefore, the risk of having more interference is less. Due to elimination of the AC process in the DAD-Less MIPv6, there will be no signal resource or power wasted. The Ping-pong movement will also have less impact on the DAD-Less MIPv6, because the DAD-Less MIPv6 follows the standard Movement Detection process. For more details regarding to the Movement Detection process, refer to chapter 2. At last, the improvement of DAD-Less MIPv6 is regardless of the movement speed of the MN, it will always reduce the handover by the duration of the DAD process time. In addition, because the DAD-Less MIPv6 has simplified the handover process rather than adding extra features to the process, the mechanism is easier to be programmed than the FMIPv6.

3.4.2 PMIPv6 and DAD-Less MIPv6

PMIPv6 is a network based mobility management protocol. Like the DAD-Less MIPv6, the PMIPv6 completely removes the AC process in a handover process while the MNs move within the same PMIPv6 domain. This is achieved by allowing the MNs use their own HoA in the PMIPv6 domain. In a PMIPv6 domain, every AR or MSG in its context acquires the HAs' information of MNs, and emulates the HAs of MNs. Therefore, in the perspective of the MNs, they have never left their Home Network. To achieve this, the protocol requires adding two extra entities to MIPv6. Although the new entities are essentially only ARs with extra function, the implementation and deployment can be more complicated than MIPv6. Another issue is that the protocol has not been fully defined yet. For example, the protocol has not clearly defined the Movement Detection process which claims not requiring any involvement of

MNs. The protocol also does not specify the size of a domain. The larger domain size is the more effective the PMIPv6 will be. However, the larger domain also has higher requirements for the LMAs, and the deployment and maintenance cost will rise. If the domain size is too small, PMIPv6 will not be effective enough, since the cross domain movement still needs to follow the MIPv6 standard. The overall improvements on the delay and packet loss during a handover are very limited in this case.

The DAD-Less MIPv6 assumes only minor modifications of MIPv6 which allow sharing many attribute as the standard MIPv6. For example, the DAD-Less MIPv6 does not need to redefine a new Movement Detection process as the PMIPv6. Without introducing any additional entities in the DAD-Less MIPv6, the proposal is easier to be implemented and deployed than the PMIPv6. Consequently, it reduces much risk for ISPs. The DAD-Less MIPv6 is also a global protocol as the standard MIPv6. Without defining domains, MNs can still have smaller handover latency and lower packet loss.

3.5 Combining DAD-Less MIPv6 with existing proposals

In the beginning of designing the DAD-Less MIPv6, the proposal was considered to be a complement to other existing proposals rather than a contender. The DAD-Less MIPv6 can be used solely to improve MIPv6, and it can also be used in conjunction with other handover protocols. This section describes the required changes to the other existing proposals when DAD-Less MIPv6 is used with them together.

3.5.1 DAD-Less MIPv6 and FMIPv6

The FMIPv6 shortens the latency in a MIPv6 handover process, and it reduces the packet loss by forwarding the buffered packets from the OAR to the NAR. The DAD-

less MIPv6 removes the AC process completely in a MIPv6 handover process. Combining FMIPv6 and DAD-Less MIPv6 allows a MIPv6 handover to avoid the AC process and lower the packet loss.

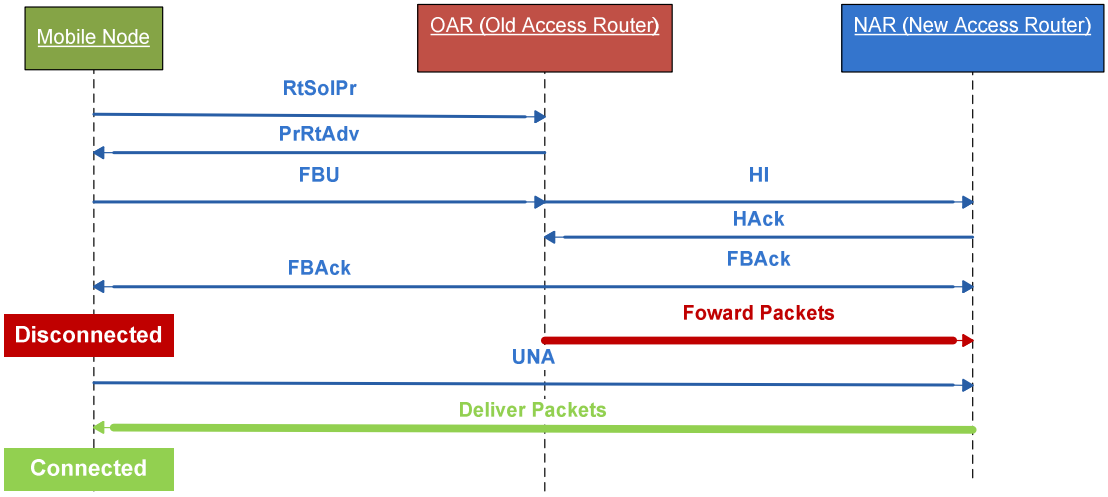

Figure 3.4 Predictive mode of FDAD-Less MIPv6

Figure 3.4 demonstrates a handover processes by using the combination of FMIPv6 and DAD-Less MIPv6 (FDAD-Less MIPv6). In fact, from the surface, there no difference in the handover process between FDAD-Less MIPv6 and FMIPv6. However, the content of the messages have been modified.

The changes within the messages in the predictive mode are:

a. The FBU message will no longer contain the CoA of an MN. It only needs the HoA and the MAC address of the MN.

b. Once the NAR receives a HI message, the NAR will reply with a HAck message immediately after the HoA and MAC address of the MN have been stored to its routing table. The difference here is that the NAR no longer need to wait for the completion of the DAD process. This allows the FDAD-Less MIPv6 to perform faster than the FMIPv6. Therefore, with the same amount of overlap area be-

tween two ARs, there will be less chance for FDAD-Less MIPv6 to switch to the reactive mode.

c. Hack and FBack are sent only for confirming the registration of MN on the nAR is successful. There is no CoA included in both messages.

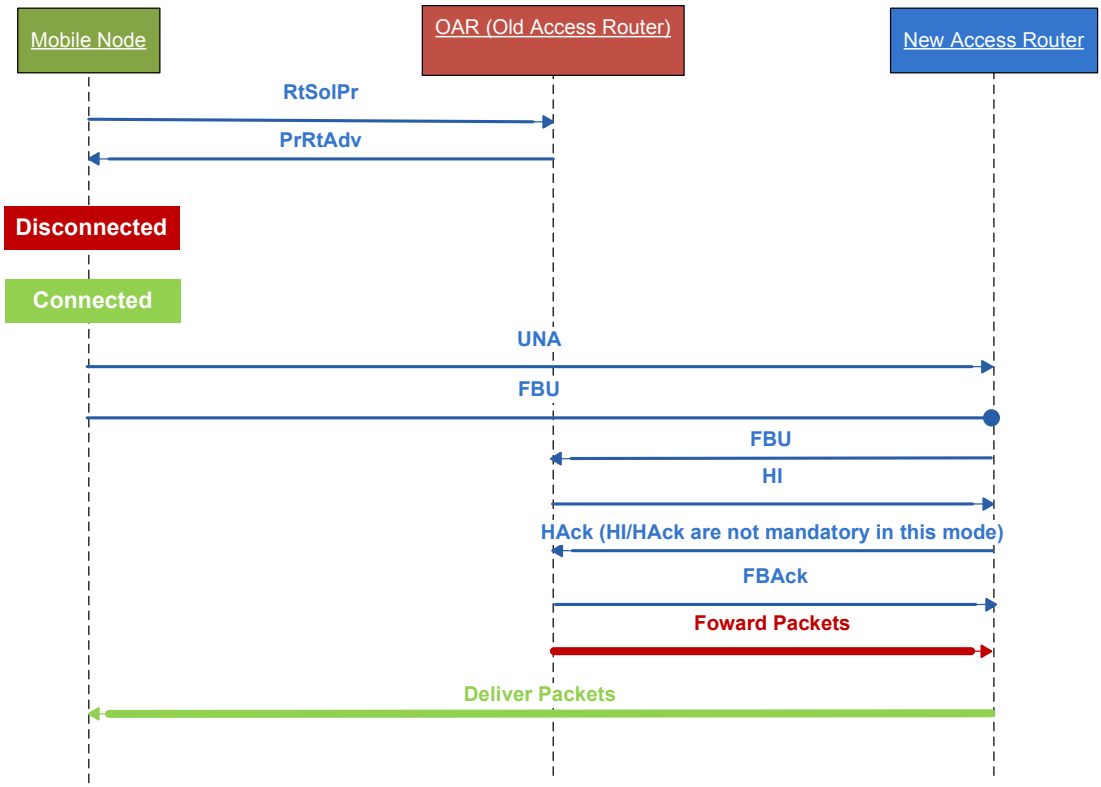

Figure 3.5 Reactive mode of FDAD-Less MIPv6

In the reactive mode, the changes are not obvious in the sequence of the messages either. The content of the messages are modified in the same way in the predictive mode. Performing the AC process before or after losing the connection between the MNs and the OARs is one of the main differences between the predictive mode and the reactive mode. It is also the reason why the reactive mode contains greater delay than the predictive mode. Since the AC process is completely removed in this combination of the FMIPv6 and the DAD-Less MIPv6, and the key improvement of the

predictive mode is performing the AC process earlier than it is required, the delay difference between these two modes are almost eliminated. However, the combination of the FMIPv6 and the DAD-Less MIPv6 takes the advantage of the FMIPv6. By using buffers, the packet loss during a handover can be minimized.

3.5.2 DAD-Less MIPv6 and HMIPv6

The HMIPv6 shortens the handover delay by placing a closer proxy server (MAP) to reduce the BU process delay. To allow the HMIPv6 to work with the DAD-less MIPV6, the LCoA and the RCoA are removed. Figure 3.6 below demonstrates the difference between normal HMIPv6 and the combination of HMIPv6 and DAD-Less MIPv6.

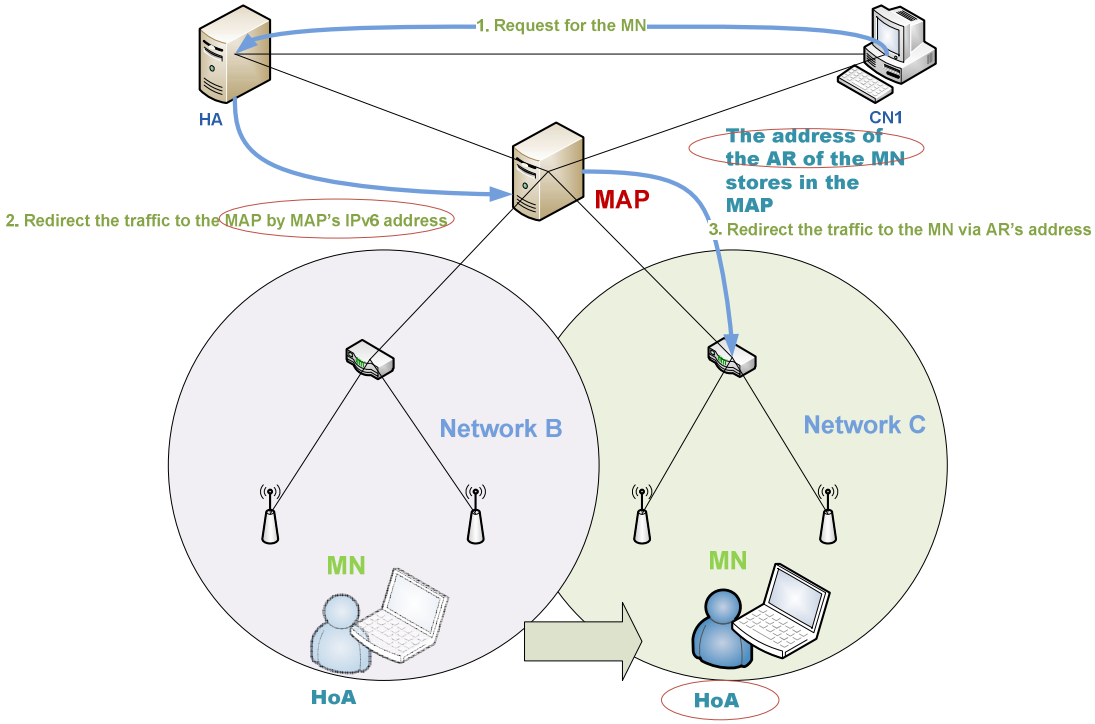

Figure 3.6 The HMIPv6 handover process

Like Figure 3.6 illustrated, there are no LCoA and RCoA being used any more. Instead, the routing is completed by using MAP's IPv6 address and the current AR's IPv6 address. Like the DAD-Less MIPv6, the MN is using its HoA all the time. Due

to this change, the content of the BU messages are also modified. The BU messages between an MN and a MAP includes the address of the MN's current AR rather than its LCoA. The BU messages between a MAP and a HA includes the address of the MAP rather than the MN's RCoA.

3.5.3 DAD-Less MIPv6 and FHMIPv6

Similar to using the combination of DAD-Less MIPv6 and FMIPv6, there is no change required in the sequence of messages in a handover when the combination of DAD-Less MIPv6 and FHMIPv6 is used. The required modifications are made in the messages, and they have been explained in the last two sections separately.

3.5.4 DAD-Less MIPv6 and PMIPv6

The DAD-Less MIPv6 and the PMIPv6 both intend to eliminate the usage of CoA in the mobility management which make them very similar. The approach and scale of these two mechanisms are very different. However, the concept of the PMIPv6 is still very similar to the combination of the DAD-Less MIPv6 and the HMIPv6. The combination of the DAD-Less MIPv6 and the PMIPv6 is more valuable for the inter-domain handovers.

The standard PMIPv6 is unable to improve the handover process when there is an inter-domain handover. The inter-domain handover follows the MIPv6 protocol, so the handover latency is same as MIPv6 too. By combining the DAD-Less MIPv6 and the PMIPv6, the inter-domain handover will benefit from the advantage of the DAD-Less MIPv6. There are no obvious changes required for PMIPv6 and DAD-Less MIPv6 to use them together. For more detail regarding to integrating the MIPv6 and the PMIPv6, please refer to [5].

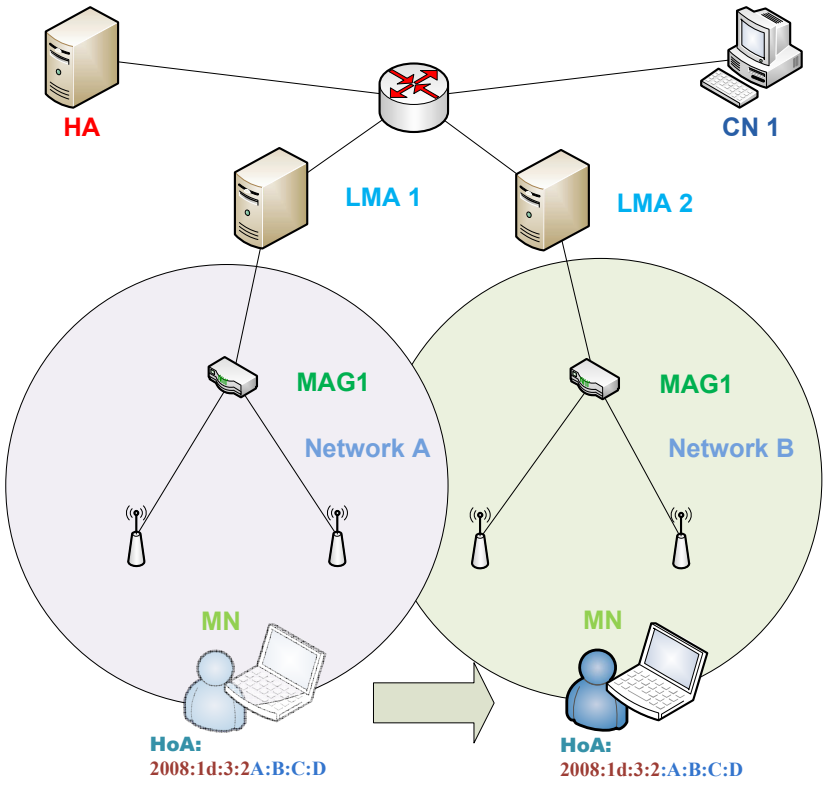

Figure 3.6 The combination of the HMIPv6 and the DAD-Less MIPv6 handover process

3.6 Simulating the DAD-Less MIPv6 with the existing proposals

As mentioned in Chapter 2, there are no reliable simulations models available for FMIPv6, HMIPv6 and PMIPv6 till the completion of this Masters thesis. To design their reliable simulation models would require more time than it is available for Masters study. Therefore, in the following chapter, only the DAD-Less MIPv6 and the standard MIPv6 are simulated and compared to each other, leaving full exhaustive comparative evaluation of all handover solutions for MIPv6 for future studies.

3.7 Possible impact

The DAD-Less MIPv6 takes the advantage of extreme of the large number of IPv6 addresses to shorten the MIPv6 handover process. Since all MNs will use their HoA in any access network, it may change the hierarchical structure of Internet. The struc-

ture of Internet is considered to be affected in small scale when MNs are only the end users. However, if an MN is mobile router with multiple other mobile routers below its hierarchy, the impact to Internet structure can be enlarged. However, so far or even may be for a long time in future, the wireless technologies does not seem to be as stable and capable as the wire technologies. Therefore, there are no obvious advantages for using multi layers of mobile routers to provide Internet connection for ISPs. This situation may be very rare.

3.8 Extra

3.8.1 Ping-pong movement

There are many research publications trying to resolve the problems caused by the Ping-pong movement. Usually a ping-pong movement refers to a back and forth movement. Figure 3.7 depicts ping-pong movement at the overlap area of two access network. The most common solution for such scenario includes complicated movement pattern detection mechanisms.

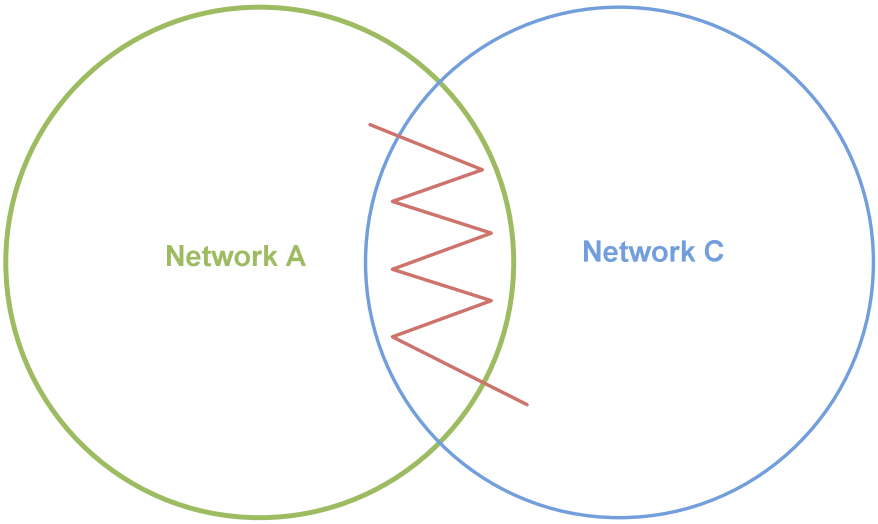

Figure 3.7 An example of a Ping-pong movement

However, in practice, the movement pattern detection is hard to be accurate with shadowing and noise in the wireless environment. The introduction of new equipment may have high cost in both implementation and installation.

In general, the ping-pong movement has great impact when soft handover or anticipated handover mechanism is used. Therefore, as long as a hard handover is too short to affect the QoS, it is not necessary to use soft handover or anticipated handover. This may be also the reason why soft handover or anticipated handover are not used in mobile phone networks. In addition, the impact of Ping-pong movement can be controlled by well planned ARs's placement. If the location of each AR is considered with geographical conditions, the possibility of Ping-pong movement can be much reduced. An extreme case of such is providing wireless connections to trains. Furthermore, if the Ping-pong movement starts to affect the QoS for a walking user, the user would stop the movement pattern to avoid the QoS degradation. A user using VoIP walking back and forth in an overlap area is a typical example of such. In conclusion, finding effective solution of the Ping-pong movement should not aim on reducing the impact caused by the movement, but at reducing the occurrence of such movement.

3.8.2 Other proposed solutions

During the process of the repeated investigations, two other solutions were proposed. One of them is very similar to SMIP, and the other one improves FMIPv6 by trying to send PrRtSol and PrRtAdv much earlier than it is required with assist of Media Independent Handover (MIH, IEEE 802.21) standard. However, both proposals have been obsolete due to publication of earlier research results suggesting the same approaches [44], [55].

3.9 Summary

In the beginning of this chapter, we discuss further possible improvements in design of a MIPv6 handover by analysing each sub-process and other existing proposals. A new solution, DAD-Less MIPv6 which removes the AC process from a MIPv6 handover completely is presented after the analysis. The solution is different from existing proposals [1], [26], [53]. Next, the mechanism of DAD-Less MIPv6, the feasibility of the solution, and the comparisons of the solution to the existing proposals, and the method and benefit of combining the solution with the existing proposals are also described. The possible impact of DAD-Less MIPv6 to Internet is discussed, and followed with extra discussions. The next chapter will focus on describing the assumption of the simulations for both the standard MIPv6 and DAD-Less MIPv6, and comparing their performance together.

Chapter 4 Simulation Model, Simulations and Results

Last chapter has described the details of the DAD-Less MIPv6 which is a proposed mechanism in this thesis. In the academic world, to verify the performance of such a mechanism, we can conduct simulations and/or experiments. In this case, we choose to use simulations. In this chapter, we will describe the main assumptions of our simulation studies and the simulation models used to evaluate performance of the DAD-Less MIPv6 and the standard MIPv6. The numerical results obtained from simulations are presented and discussed in the later part of this chapter.

4.1 Assumptions and simulation models in INET framework over OMNeT++

Our goal of this chapter is to compare the performance of the standard MPv6 handover mechanism with the DAD-Less MIPv6. To make it possible, we have made the following assumptions.

Assumption 1 (simulated environment): The performance of the standard MPv6 handover mechanism and the DAD-Less MIPv6 is done in a flat terrain, considering a rectangle of 850 meters x 850 meters. There are two base stations in this rectangle, represented by (i) the Access Point of the Home Agent and (ii) the Access Point of the Foreign Access Router. Assuming the Cartesian coordinates, with the point (0,0) in the lower left corner of this rectangle , these two Access Points (APs) are located at the point (249, 172) and (556, 172), respectively. For demonstration, refer to Figure 4.1.

The assumed terrain and its size have been assumed in the publications [7]. Note that the locations of APs within the simulated rectangle can be arbitrary as long as the Mobile Node (MN) is within the access range of an AP after it moves out from the access range of another AP.

Assumption 2 (behaviour of the user): The analysis of the handover mechanisms is studied from the point of a single user, represented by a single MN. The initial location of this MN has been selected at the point (180, 100). This MN is moving at the speed of v=1 m/s, from West to East, along a straight line from the Home Network towards to the Foreign Access Network. It is assumed that the user who is using this MN is engaged in watching a video, or follows a direction indicated by "street view" shown on Google Maps. The simulated UDP video traffic stream starts at t_0=200 seconds, and the size of the video is 20 Mega Bytes. This video streaming uses packets of size of 1Kbytes and has 10 milliseconds waiting interval.

The assumed movement pattern can be considered as a typical movement pattern for a pedestrian. It has been used in multiple publications, including [16], [44], [45], [7] and [55], so by assuming that same model of mobility we make of our results comparable with those published in [7]. The locations assumed for both APs, as well as the initial location of the MN and its speed can be arbitrarily changed, as long as the MN stays within the access range of an AP after it moves out from the access range of another AP. If the MN enters a new Access Network, it needs to stay within this Access Network for at least the duration of a handover. Otherwise, the handover would never be completed, and the MN would lose its Internet connectivity. These limitations have been taken into account when selecting the speed of MN's movement, its initial loca-

tion and the locations of both APs, within the simulated area specified by Assumption 1. Note that the type and parameters of data traffic could be arbitrarily selected too, as long as it will lead to the need for a handover during a simulation. Otherwise, we would not be able to assess its performance.

Assumption 3 (Networking characteristics): The transmission range of every node is 200 meters. The transmission range is determined by the power of the transmitters, the sensitivity of the receivers, the thermal noise, the path loss coefficient, and the SNIR threshold. More details relating to these parameters will be presented in Assumption 4. The Access Routers transmit at the rate of 100 Mbps.

Both the transmission range and the transmission rates are set to be the same as [7] Note that the transmission rate of routers can be arbitrarily selected as long as it is sufficiently large for not incurring any packet losses. Otherwise, the packet losses observed during simulation can be contributed not only by handover processes. Note too that the assumed values have no effect on the duration of handover processes. Since all these parameters are in agreement with [7] , our simulation results should be comparable to their results.

Assumption 4 (Radio transmission characteristics): The following radio transmission characteristics of the simulated wireless network are assumed: the power of the transmitters equals to 2.0mW, the sensitivity of the receiver equals to -82mW, the thermal noise equals to -110dBm, the path loss coefficient equals to 2, and the SNIR threshold equals to 4dB.

The power of the transmitters specifies the amount of power of radio frequency energy that the transmitter produces at its output, [59]. The sensitivity of the receiver indicates how faint a radio frequency signal can be successfully received by the receiver, [60]. The thermal noise is the electronic noise generated by the thermal agitation of the change carriers inside an electrical conductor at equilibrium. The thermal noise power level is approximate -110 dBm if the carrier bandwidth is 2.4 MHz which one of the WiFi operating bandwidth. The pass loss coefficient defines the loss of a signal that encounters inside a building or densely populated areas over distance, [61]. Setting the value to 2 indicates a vacuum or infinite space. SNIR threshold is the signal to noise ratio which can also be written as SNR threshold, [62]. The assumed parameters' setting can be considered as typical, and it represents the default values of OMNeT++ [35]. These settings also agree with [7].

Assumptions 1-4 have been implemented in our simulations by using models developed in INET Framework over OMNeT++; see Appendix A. Since the main focus of this research is to reduce the latency during a handover phase in the wireless MIPv6 networks, the simulation scenarios have been also focused on acquiring the latency data during a handover process. Figure 4.1 depicts the simulated scenario in which an MN (user) walks along a straight street while watching a video clip.

As stated in Assumption 1, the simulated environment represents an 850 m x 850 m rectangle with WiFi wireless Internet connections. There are seven nodes in total in this area. They are: an MN, a hub, two Correspondent Nodes, two APs and three routers. The MN is here represented by a laptop icon (the top left of the figure) and it is labelled as "MN[0]". The routers are indicated by common router icons which are shown as short blue cylinders. Two Correspondent Nodes are presented by the desktop images. Among the three routers, the router on the left hand side of the figure is

the Home Agent of the Mobile Node which is labelled as “Home_Agent” underneath. The router on the right hand side of the figure is the Foreign Network Access Router of the MN, labelled as “R_1”. Both the Home Agent and the Access Router are connected by a single access point. The circles in Figure 4.1 indicate the transmitting and receiving ranges of the WiFi connections. Now, let us explain the parameters and settings for each node in detail.

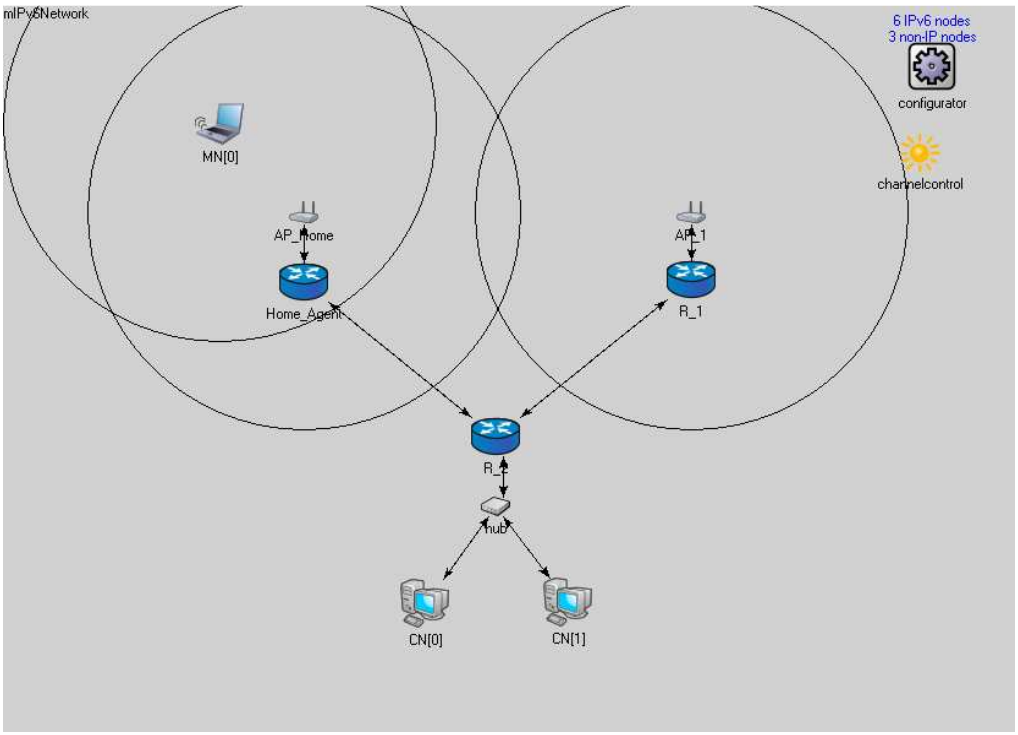

Figure 4.1 the simulation scenario

MN[0]

As mentioned in the assumptions, we have simulated an MN user assuming that he/she is watching a video clip while walking along a street. The node MN[0] in Figure 4.1 represents the simulated MN user. The behaviour of MN[0] is set by the lines of code in an “.ini” file, which is the simulation configuration file for OMNeT++ simulation environment; see Code 4.1.

```
# mobility
1 **.MN[0].mobilityType = "RectangleMobility"
2 **.MN[0].mobility.debug = false
3 **.MN[0].mobility.x1 = 180
4 **.MN[0].mobility.y1 = 100
5 **.MN[0].mobility.x2 = 530
6 **.MN[0].mobility.y2 = 110
7 **.MN[0].mobility.startPos = 0
8 **.MN[0].mobility.speed = 1mps
9 **.MN[0].mobility.updateInterval = 0.1s
# UDP application setting
10 **.MN[0].numUdpApps = 1
11 **.MN[0].udpAppType = "UDPVideoStreamCli"
12 **.MN[0].udpApp[*].serverAddress = "CN[0]"
13 **.MN[0].udpApp[*].localPort = 9999
14 **.MN[0].udpApp[*].serverPort = 3088
15 **.MN[0].udpApp[*].startTime = 200s
```

Code 4.1Parameters for MN[0] in the configuration file of OMNeT++.

The first part of Code 4.1 is the mobility setting for MN[0]. According to the first line of the part, MN[0] has been assigned with a "RectangleMobility" model. This means MN[0] moves in a rectangular path, the path is defined by the diagonal points of the rectangle. In this case, they are (180, 100) and (530, 110) which are specified from line 3 to line 6. This model is used here for simulating a straight line movement. It is achieved by ending the simulation before MN[0] starts to turn. Moreover, this model can be used to simulate the ping-pong movement which may be evaluated in future study. Line 7 indicates the starting point of MN[0] in the rectangular path, and it has been denoted as 0. The speed of MN[0] is set to 1 metre per second. The position of MN[0] in the animation will be updated every 0.1 second which is irrelevant to the

result of the simulations. The second part of Code 4.1 is related with the UDP application which is running on MN[0]. The line 10 and 11 specifies one UDP video stream client running on MN[0] . The line 12 to 14 defines the address of the UDP video stream server, the port used on MN[0] for transmitting UDP traffic and the UDP traffic port at the server side. The last line of Code 4.1 sets the UDP traffic starts at 200 seconds of the simulation.

CN[0]

There are two Correspondent Nodes in Figure 4.1, but only CN[0] actually communicates with MN[0]. CN[0]-related definitions are listed in Code 4.2.

```
# UDP application setting
1 **.CN[0].numUdpApps = 1
2 **.CN[0].udpAppType = "UDPVideoStreamSvr"
3 **.CN[0].udpApp[*].videoSize = 20MB
4 **.CN[0].udpApp[*].serverPort = 3088
5 **.CN[0].udpApp[*].waitInterval = 10ms
6 **.CN[0].udpApp[*].packetLen = 1000B
```

Code 4.2 Parameters for CN[0] in the configuration file on OMNeT++.

In Code 4.2, line 1 indicates the CN[0] has only 1 UDP application running, and line 2 specifies the type of UDP application is a video stream server. Line 3 defines the size of the video which is transmitted from CN[0] to MN[0], and it is set to 20 MBytes in this simulation scenario. Line 4 specifies the port used for the UDP traffic stream which is 3088, and it corresponds with the definition in Code 4.1 line 14 for MN[0]. The waiting interval of each UDP packet is set to be 10 milliseconds by line 5. Line 6 defines the size of each UDP packet to 1 Kbyte. By knowing the size of the

video, the wait interval and the packet size, we can derive that the required time for completing the transmission of the whole video is 200 seconds.

Home_Agent, R_1 and R_2

The Home_Agent, R_1 and R_2 are the routers shown in Figure 4.1. Their attributes as defined in OMNeT++ are shown in Code 4.3.

```
# Ethernet NIC configuration
1 **.eth[*].queueType = "DropTailQueue" # in routers
2 **.eth[*].queue.frameCapacity = 10  # in routers
3 **.eth[*].mac.promiscuous = false
4 **.eth[*].mac.address = "auto"
5 **.eth*.mac.txrate = 100Mbps
```

Code 4.3 parameters for routers in the configuration file in OMNeT++.

In Code 4.3, line 1 defines the queue type used in all three routers, which in all three cases is "Drop Tail Queue"[1]. Line 2 defines the maximum size of the queue inside of the routers, and line 3 set the routers only accept the packets that matches the Media Access Control (MAC) address. Line 4 specifies that the router's MAC addresses will be random generated by OMNeT++ simulation environment. Line 5 defines the transmission capability of the routers to 100 Mbps.

The definitions of these nodes are written in the C++ files and NED (Network Description) files. The C++ files define the behaviour of objects for different simulation models. A NED file can combine the simulation models defined in C++ files to a new

[1] Drop Tail Queue, is a simple buffer management algorithm used by Internet routers to decide when to drop packets. In contrast to the more complex algorithms like RED and WRED, in Drop Tail Queue the data traffic is not differentiated. Each packet is treated identically: when the buffer is filled to its maximum capacity, the newly arriving packets are dropped until the buffer has enough room to accept incoming traffic.

simulation model. “Home_Agent”, “R_1” and “R_2” are simulation models which are described by the NED files. The files are placed under the folder “nodes” which is shown in Figure 4.1. Since the details of the implementation of these nodes are not the focus of this section, only an example of the definition of the “Home_Agent” simulation model is shown in Figure 4.2 and Figure 4.3.

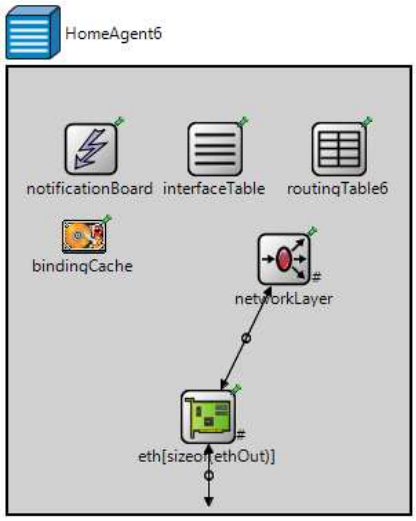

Figure 4.2 The HomeAgent6 simulation model defined by a NED file

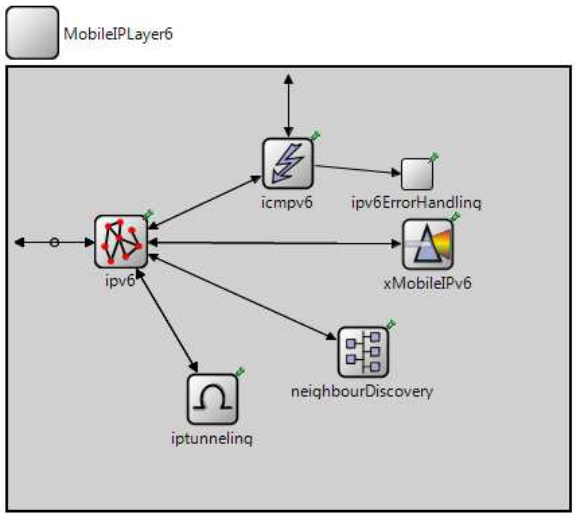

Figure 4.3 The networkLayer in the Home Agent simulation model

“Home_Agent” node in is an instance of the simulation model of the HomeAgent6 which is shown in Figure 4.2. It contains such objects as “networkLayer”, an

"eth[sizeof(ethOut)]", a "notificationBoard", an "interfaceTable", "routingTable6" l and a "bindingCache". Their properties are as follows:

(a) "networkLayer" contains the functions of MIPv6;

(b) "eth[sizeof(ethOut)]" is the Ethernet model which contains functions relate to Ethernet;

(c) "notificationBoard" is used for catching network changes which are mainly used for programming purpose;

(d) "interfaceTable" stores the information of the Ethernet interfaces;

(e) "routingTable6" is where the Access Router stores routes; and

(f) "bindingCache" is where keeps the network location of Mobile Nodes.

Figure 4.3 shows the customised network layer for the HomeAgent6 simulation model. Its components are : "IPv6", "icmpv6" , "ipv6ErrorHandling" , "iptunneling" and "xMIPv6". They the components of a MIPv6 network model. "R_1" and "R_2" in Figure 4.3 are the instances of a Router6 simulation model. The difference between the HomeAgent6 and the Router6 is the extra entity "xMIPv6" in the network layer, as demonstrated in Figure 4.3.

"AP_Home" and "AP_1"

"AP_Home" and "AP_1" are the access points which are connect with "Home_Agent" and "R_1" respectively, as shown in Figure 4.1. The properties of "AP_Home" and "AP_1" are defined in Code 4.2.

```
# ALL APs common parameters
1 **.AP*.wlan.mgmt.beaconInterval = 0.1s

# Access Point AP_Home ; AP_1
```

```
2 **.AP_Home.wlan.mgmt.ssid = "HOME"
3 **.AP_Home.wlan.mac.address = "10:AA:00:00:00:01"
4 **.AP_Home.eth[0].address = "10:AE:00:00:00:02"
5 **.AP_Home.eth[0].txrate = 100Mbps

6 **.AP_1.wlan.mgmt.ssid = "AP1"
7 **.AP_1.wlan.mac.address = "10:AA:00:00:A1:01" #the A1:01 specifies AP_1:interface 1
8 **.AP_1.eth[0].address = "10:AE:00:00:A1:02" #the A1:02 specifies AP_1:interface 2
9 **.AP_1.eth[0].txrate = 100Mbps
```

Code 4.4 Parameters for Access Points in the configuration file

Line 1 defines the interval of the transmission between each beacon frame for all the Access Points in the simulation. A beacon frame is one of the management frames in IEEE 802 based WLANs, and it contains the capability of the device or network, [2]. For IEEE 802.11 standards, the default value is 0.1 second, [14] and [16] and. Line 2 and 6 set the Service Set Identifier (SSID) for "AP_Home" and "AP_1", respectively. The line 3, 4, 7 and 8 defines the Media Access Control (MAC) addresses for the network interfaces that are on "AP_Home" and "AP_1". The line 5 and 9 defines the transmission capability of the device which is 100 Mega bits per second. Again, the transmission capability is set to high values here for elimination of delays which may be caused by the network rather than a handover.

Ethernet lines and wireless environment

The properties of the Ethernet lines and the wireless environment are remained to be default value which are also the values used in [7]. The propagation delay can be almost neglected since they are set to be 1×10^{-6} seconds with 512×10^{6}; this can also help us to obtain accurate results in the duration of a handover.

Switching between MIPv6 simulation and DAD-less MIPv6 simulation

To switch between the standard MIPv6 and the DAD-Less MIPv6, one needs to change a single value in the configuration. Namely, one should set:

.*.networkLayer.dadlessMipv6Support = **true

As stated earlier, in our simulated scenario, the MN is moving from its Home Network to a new Access Network with the speed of 1 m/s. At time t= 200 seconds, the MN begins a UDP video streaming. The handover process is initiated at about 230 seconds of the simulated time, with uncertainties caused by the random processes occurring in the physical and link layer. The simulation ends after 250 seconds of the simulated time.

The key measurements taken in the scenario are the overall handover delay, the L2 handover delay and the L3 handover delay. The overall handover delay consists of the delay caused from all layers which include also the transport and the application layers. This delay is the actual delay that the user would experience during a handover.

4.2 Numerical Results

As documented in [39], the final results from stochastic discrete-event simulation reported without statistical errors cannot be considered as being credible. The simulations reported in this Master s thesis have involved such stochastic phenomena as (i) the duration of the Layer 2 handover, (ii) the interval of the Router Advertisement messages, and (iii) the duration of the DAD process. The interval of the Router Advertisement messages and the duration of the DAD process are both defined in RFC 3775. They are described in details in Chapter 2. All these three specific instances are determined by random numbers generated by a pseudo-random number generator.

The generator used by OMNeT++ is known as the Mersenne Twister Random Number Generator [31]. Traditionally, for analysis of statistical errors, samples of output data of fixed size are collected. However, such an approach depends on pure luck, as nobody can guess in advance how large sample is needed to obtain the final results with small statistical error. The alternative solution is known as sequential scenario of simulation or simply, sequential simulation. It is generally accepted as the only approach that allows controlling the accuracy of the final simulation results. According to this concept, the simulation continues until the statistical error of results drops below an acceptable level. This can be demonstrated by Figure 4.4.

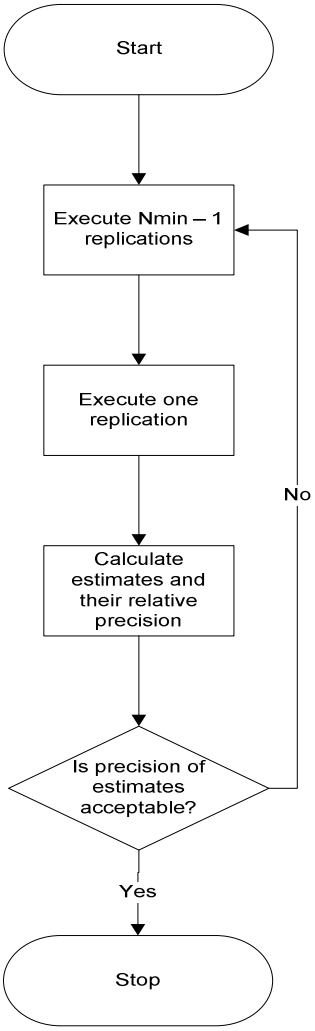

Figure 4.4 Flowchart of sequential terminating simulation

In Figure 4.4, it shows, new replications of the simulation are being executed until the precision of estimates (i.e. the statistical error) is acceptable. In our analysis of mean delays related with handovers in MIPv6 and DAD-Less MIPv6, the acceptable precision of estimates is assessed by the relative statistical error, at a given confidence level. The relative statistical error is defined as

$$\delta(N) = |\Delta(N)/\overline{D}(N)| \times 100\% \qquad (4.1)$$

where $\Delta(N)$ means the half-width of the confidence interval (or the margin of error), and $\overline{D}(N)$ is the mean of the results; i.e.

$$\Delta(N) = t_{N-1,1-\frac{\alpha}{2}} S(N)/\sqrt{N} \qquad (4.2)$$

$$\overline{D}(N) = \frac{1}{N}\sum_{i=1}^{N} d_i \qquad (4.3)$$

d_i (i=1, 2, …, N) are values of delays recorded in consecutive replications, and S(N) is the estimated standard deviation of delays, i.e.

$$S^2(N) = \frac{1}{N-1}\sum_{i=1}^{N}(d_i - \overline{D}(N))^2 \qquad (4.4)$$

All our results have been determined assuming that their maximum acceptable relative error is 1%, at the confidence level (1-alpha)=0.1. To achieve such relative error of the final results, we needed to run up to 66 independent replications.

Figure 4.5 and Figure 4.6 show the snapshots from the results obtained from our replicated simulations.

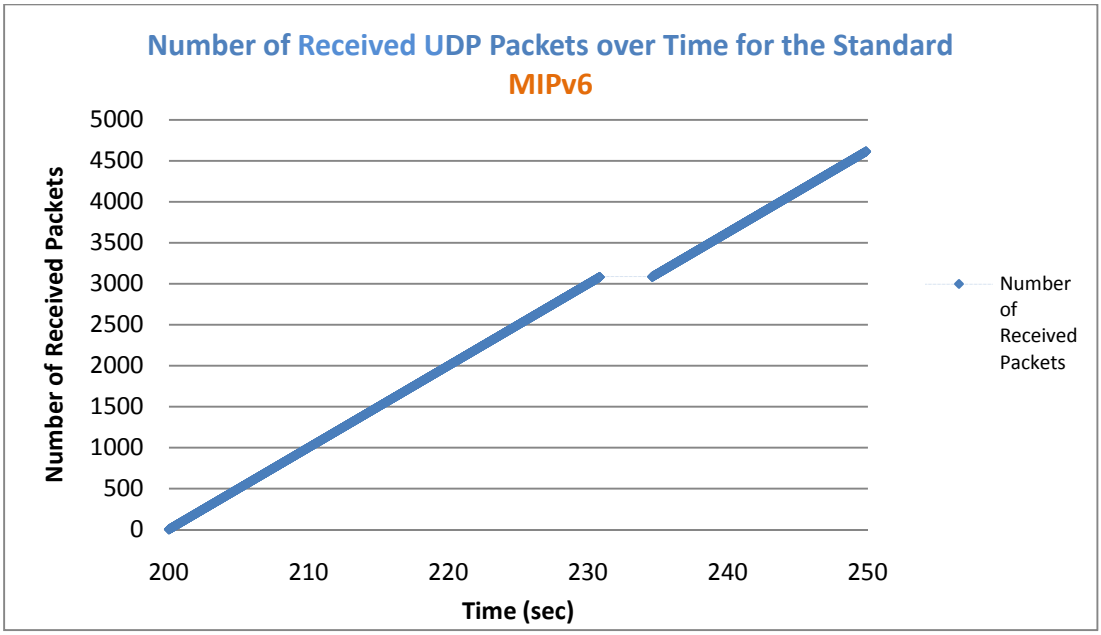

Figure 4.5 Number of received UDP packets over time during simulation of the standard MIPv6

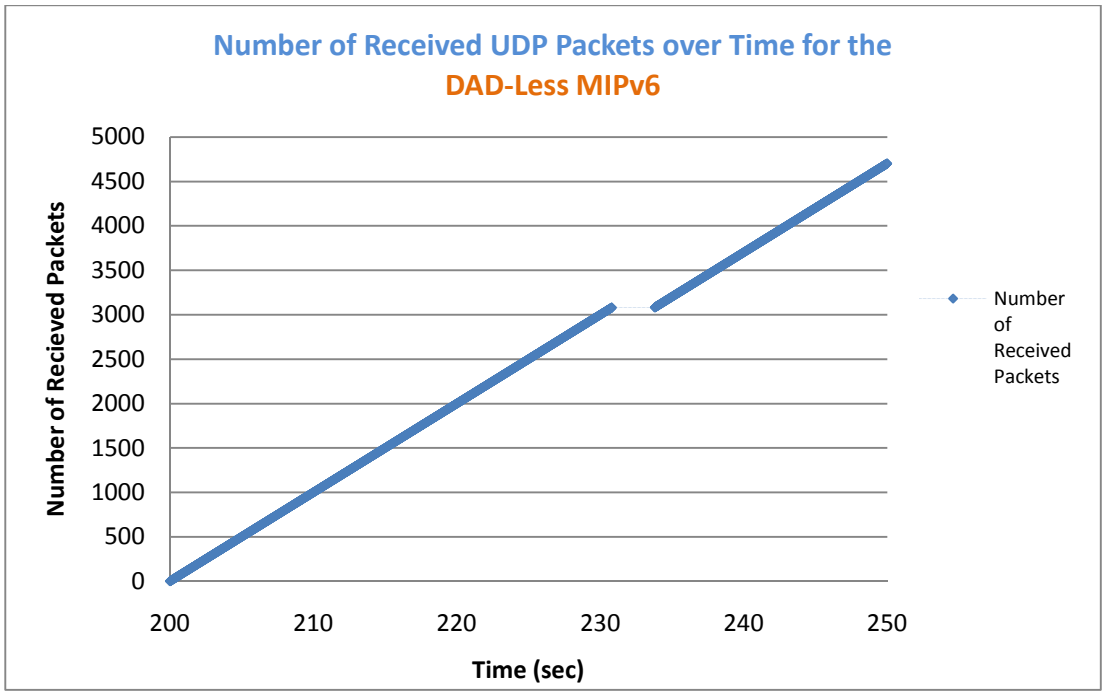

Figure 4.6 Number of received UDP packets over time during simulation of the DAD-Less MIPv6

In the figures, the X-axis indicates the running time of a given simulation, and the Y-axis indicates the recorded number of received packets. The value of the X-axis starts from 200 seconds which is the starting time for the UDP video stream traffic. Due to the large amount of data, although the figures are drawn by dotted lines, the dots ap-

pear to be a continued line, except at the time of the handover. By comparing both figures, we can observe that the handover process takes shorter time in the case of DAD-Less MIPv6.

In general a handover consists of physical layer process time, delays from Layer2, Layer3 and other higher layers, i.e.

$$D_{overall} = D_{L2} + D_{L3} + D_{other\ layer} \quad (4.5)$$

The process time from the physical layer is usually hard to be identified, because they are much lower order than the overall delays. They are usually distributed and embedded in L2, L3 and other high layers which is reason why this delay is not included in the formula above.

The L2 handover starts with a scanning request which searches for new Access Point, and it ends with associated confirmation message. The delay for L2 is the interval between these two messages. The layer 3 delay is measured by the interval between the ends of the L2 handover and the last Binding Update acknowledge message received for the MN. The delays from other layers are computed by subtracting the L2 and L3 delays from the overall delay. The overall delay is the delay between UDP packets during a handover process. By following these conditions, we have obtained the final results listed in Table 4.1.

Attribute	Value
The Overall Average Delay in standard MIPv6	3.504 ± 0.018 (smaller than 1% relative statistical error at 0.95 confidence level)
The Overall Average Delay in DAD-Less MIPv6	3.000 ± 0.016 (smaller than 1% relative statistical error at 0.95 confidence level)

Average Layer 2 Delay in standard MIPv6	0.655 ± 0.002 (smaller than 1% relative statistical error at 0.95 confidence level)
Average Layer 2 Delay in DAD-Less MIPv6	0.655 ± 0.002 (smaller than 1% relative statistical error at 0.95 confidence level)
Average Layer 3 Delay in standard MIPv6	2.605 ± 0.013 (smaller than 1% relative statistical error at 0.95 confidence level)
Average Layer 3 Delay in DAD-Less MIPv6	2.101 ± 0.011 (smaller than 1% relative statistical error at 0.95 confidence level)

Table 4.1 The final results

The results from Table 4.1 match the expectation. The difference of average overall handover delays between the standard MIPv6 and the DAD-Less MIPv6 is close to 0.5 seconds. This confirms expectations, as the average duration of the DAD process is 0.5 seconds which has been described in Chapter 3.1. The difference between the Layer 3 delay in the standard MIPv6 and in the DAD-Less MIPv6 is also 0.5 seconds. This demonstrates that the network layer handover has been improved by the DAD-Less MIPv6 handover mechanism. In the result, we have decreased the handover latency for MIPv6 by 25%.

4.3 Summary

In this chapter, we have briefly described our simulation models and simulation methodology used for obtaining results with small statistical errors. We have also presented assumptions of our simulation studies. The simulation results which we have presented prove that the handover mechanism proposed in this thesis decreases the standard MIPv6 handover latency by 25%. In principle, this mechanism should also benefit other existing proposed solutions.

Chapter 5 Conclusions

In this Masters thesis, we have proposed a modification of the MIPv6 handover process named DAD-Less MIPv6 which significantly shortens the duration of the handover of the standard version. The standard solution consists of five sub-processes which are: Movement Detection, Candidate Access Router Selection, Address Configuration, Authentication & Authorization and Binding Update. Each of these sub-processes contributes in extending the duration of the handover process. We have looked at the existing solutions and tried to make them more efficient. They include the standardized versions as well as such well-known ones as: FMIPv6, HMIPv6, FHMIPv6, SMIP and PMIPv6. We have proposed three new mechanisms for shortening the handover delay. During the course of our investigations, we have found that two of them had been unfortunately earlier proposed, see section 3.6.2 for more details. Therefore, we have focused on the third one which we call the DAD-Less MIPv6 mechanism. The feasibility study of this mechanism and its performance evaluation is the main subject of this thesis.

In the process of validating the mechanism, finding reliable simulation model is a critical issue. We have found that most of the previous publications in this area report in NS2. However, these simulation models are outdated since they were designed in 2002, while the MIPv6 standard has continued to be updated also after 2002. Because of that, most of the existing simulation models cannot be validated against real test-beds. The only newly developed MIPv6 simulation model, validated against a real test-bed, has been found in OMNeT++. This model has been used by us in the performance evaluation of MIPv6, with and without the DAD-Less MIPv6. By comparing the results, we have shown that the handover process of MIPv6 with DAD-Less

mechanism is on average 0.5 second shorter than this process in MIPv6 without this mechanism. Thus we have demonstrated a 25% improvement to the standard MIPv6 handover process.

Due to lacking of reliable simulation model prevented us from comparing our handover procedure with [53]. As we mentioned early, the current widely used simulation models has diverse level of inaccuracy. Thus, there is an urgent need for their verification. Since the result from different simulator can be so different, so the result from [1], [26], [53] are incomparable to the result presented in this thesis. Especially, [53] has been published in the final stages of the Masters study, the time constraints prevented us from comparing the performance between the DAD-Less and [53]. This is left for future work.

Having surveyed existing handover proposals, one can say that universally the best solution is unknown. Each existing proposal aims to improve different sub-processes of the MIPv6 handover process. A more comprehensive and optimised solution is likely to be a combination of these proposals. We believe that by combining the concept of HMIPv6 and FMIPv6 with DAD-Less MIPv6, we may be able to obtain much better performance than currently known.

5.1 Future work:

As mentioned in Chapter 2 and earlier section of this chapter, the MIPv6 handover process can be divided into five sub-processes. This thesis has not focused on the authentication & authorization process which is related with issues of network security. To have an optimal solution for shortening the handover process, this is certainly an issue that requires further study. In addition, the Candidate Access Router Selection is

still not standardized. There exist a number of uncertainties in this sub-process, which could be explored for further reduction of the duration of the handover. However, further studies of possible improvements of MIPv6 are not possible without the development of new valid simulation models of MIPv6 and its extensions. Thus designing these models is urgently needed.

Nevertheless, the results we have obtained seem to be sufficient to justify our DAD-Less MIPv6 mechanism to be submitted as Request For Comment to IETF, which is planned in the nearest future.

References

[1]. M. Bagnulo, I. Soto, A. Garcia-Martinez and A. Azcorra, Avoiding DAD for Improving Real-Time Communication in MIPv6 Environments, IDMS/PROMS 2002, page: 73-79

[2]. Beacon frame, http://en.wikipedia.org/wiki/Beacon_frame, [last visited 2010 06/27].

[3]. T. Camp, J. Lusth, J. Matocha and C. Perkins, Reduced cell switching in a mobile computing environment, MobiCom Aug 2000.

[4]. S. Deering and R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, December 1998.

[5]. V. Devarapalli, S. Gundavelli, K. Chowdhury, A. Muhanna, NETLMM Working Group, Internet-Draft "Proxy Mobile IPv6 and Mobile IPv6 interworking draft-devarapalli-netlmm-pmipv6-mipv6-01.txt", April 25, 2007.

[6]. D. Di Sorte, M. Femminella, L. Piacentini and G. Reali, Target access router selection in advanced mobility scenarios, Journal Computer Communications Volume 29 Issue 3, February, 2006

[7]. Y. Z. Faqir, B. Christian and C. Wietfeld, An Accurate and Extensible Mobile IPv6 (xMIPV6) Simulation Model for OMNeT++, International congerence on Simulation Tools and techniques for communications, 2008.

[8]. P. Ferguson, Network Ingress Filtering:Defeating Denial of Service Attacks which employ IP Source Address Spoofing, RFC 2827, May 2000.

[9]. D. Greg, P. Brett and N. Richard, Movement Detection Optimizations in Mobile IPv6, Networks, ICON2003. The 11th IEEE International Conference on 28 Sept.-1 Oct. 2003 On page(s): 687 – 692

[10]. S. Gundavelli, K. Leung,V. Devarapalli, K. Chowdhury and B. Patil, Proxy Mobile IPv6, RFC 5213, August, 2008

[11]. A. Hasson, N. Ventura and S. Shepstone, *Mobile IP Movement Detection Optimizations in 802.11 Wireless LANs*, 2008.

[12]. S. Hesham, Mobile IPv6: Mobility in a Wireless Internet, Book, Addison-Wesley Professional, April 15, 2004.

[13]. N. Huu-Nghia and B. Christian, Scalable Proxy Mobile IPv6 For Heterogeneous Wireless Networks, Mobility 2008

[14]. IEEE, *Part 11: Wireless LAN Medium Access Control (MAC) and hysical Layer (PHY) Specifications*, IEEE Standard 802.11, 1999.

[15]. IPv4 Address Report http://www.potaroo.net/tools/ipv4/ [last visited 2010 06/26].

[16]. J. Jakubiak and Y. Koucheryavy, Precise delay analysis for IEEE 802.11 legacy Ad-hoc networks, Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th Issue Date: 22-25 April 2007 On page(s): 2956 – 2960

[17]. D. B. Johnson, C. E. Perkins, and J. Arkko, RFC 3775 - Mobility Support in IPv6, June 2004.

[18]. L. Jong-Hyouk , C. Tai-Myoung and G.Sri, Shall We Apply Paging Technologies to Proxy Mobile IPv6?, MobiArch'08, August 22, 2008.

[19]. L. Joo-Chul and P. Jung-Soo , Fast Handover for Proxy Mobile IPv6 based on 802.1I Networks, ICATCT, Feb, 2008.

[20]. K. Ju-Eun , Kum. Dong-Won , L. Yang , and C. You-Ze, Seamless Handover Scheme for Proxy Mobile IPv6, DOI 10.1109, WiMob, 2008.

[21]. L. Jun and F. Xiaoming , Evaluating the Benefits of Introducing PMIPv6 for Localized Mobility Management, Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International Issue Date: 6-8 Aug. 2008 On page(s): 74 - 80.

[22]. L. Kang-won, S. Won-Kyeong , K. Dong-Won, and C. You-Ze , Global Mobility Management Scheme with Interworking between PMIPv6 and MIPv6,WiMob, 2008.

[23]. R. Koodli, Mobile IPv6 Fast Handovers, IETF RFC 5568, 2009

[24]. J.Lai, Y. Ahmet.Sekercioglu, N.Jorda and A.Pitsillides, Performace Evaluation of Mobile IPv6 Handover Extensions in an IEEE 802.11b Wireless Network Environment, ISCC 2006, page: 161 -166

[25]. J. Lee and S. Ahn, Internet Draft I-FHMIPv6: A Novel FMIPv6 and HMIPv6 Integration Mechanism, June 2006

[26]. L. Lei, Fast Handover Using Explicit Multicast for IPv6-based Wireless LAN Networks, Doctor of Philosophy thesis, 2005.

[27]. M. Liebsch, A. Singh, H. Chaskar, D. Funato and E. Shim, (Experimental) Candidate Access Router Discovery (CARD), IETF RFC4066, 2005

[28]. N. Luke , A comparison of mobile IP handoff mechanisms, 6th Twente Student Conference on IT, Enschede, 2nd february, 2007.

[29]. P.Man Kyu, L. Jae Yong, K. Byung Chul, K. Dae Young, Design of Fast Handover Mechanism for Multiple Interfaces Mobile IPv6, IEEE, 2008.

[30]. D. Martin, D. Martin, P. Theo, E. Chris and D. Martin, Mobile IPv6 Handovers: Performance Analysis and Evaluation, 6NET Report, 2005.

[31]. M. Matsumoto and T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modelling and Computer Simulation 1998, vol.8.no.1, 3–30.

[32]. T. Narten and R. Draves, IETF RFC - 3041Privacy Extensions for Stateless Address Autoconfiguration, 2001

[33]. T. Narten, E. Nordmark, and W. Simpson, Neighbour Discovery for IP version 6, IETF RFC 2461, Dec. 1998.

[34]. D.Nystedt, NTT DoCoMo to Launch LTE Mobile Broadband in 2010. Available from: http://www.pcworld.com/businesscenter/article/154069/ntt_docomo_to_launch_lte_mobile_broadband_in_2010.html [last visited 2010 06/26].

[35]. OMNeT++ Community, http://www.omnetpp.org/ [last visited 2010 06/27]

[36]. OPNET Research http://www.tech.plym.ac.uk/see/research/cdma/Projects/opnet.htm [last visited 2010 06/27]

[37]. K. Päivi, *Candidate Access Router Discovery* 2003.

[38]. B. Park, S. Lee and H. Latchman, *A Fast Neighbor Discovery and DAD Scheme for Fast Handover in Mobile IPv6 Networks* 2006.

[39]. K .Pawlikowski, H.-D.J. Jeong. and R.J-S. Lee, *On Credibility of Simulation studies of Telecommunication Networks*, IEEE Comms Magazine, vol.40. no.1, 132-9, Jan. 2002.

[40]. T.Hain, *A Pragmat Report on IPv4 Address Space Consumption.* Available from: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_8-3/ipv4.html [last visited 2010 06/26].

[41]. K. Pyung-Soo, K. Sang-Eon, J. JongSam, and L. Seong-Choon, Proactive Correspondent Registration for Proxy Mobile IPv6 Route Optimization, IJCSNS International Journal of Computer Science and Network Security, vol.7. no.11, November, 2007.

[42]. K. S. Rajeev and P. E. Charles, Mobile Inter-networking with IPv6: Concepts, Principles and Practices, Book, Wiley-Intterscience, July 9, 2007.

[43]. T. S. Rappaport, *Wireless communications principles and practices*, Book, Prentice-Hall, 2002.

[44]. H. Robert , Z. Zhe Guang , S. Aruna , *S-MIP: A Seamless Handoff Architecture for Mobile IP*, IEEE INFOCOM, 2003.

[45]. H. Robert and S. Aruna , *A Comparison of Mechanisms for Improving Mobile IP Handoff Latency for End-to-End TCP*, MobiCom 2003, September, page:14-19.

[46]. L. Ruidong , L. Jie , W. Kui , X. Yang ,and X. Jiang , An Enhanced Fast Handover with Low Latency for Mobile IPv6, IEEE, August, 2006.

[47]. H. Shariq and F. Ahmad, *Handoff latency analysis of mobile IPv6 protocol variations*, Computer Communications vol.30, 849–855, 2007.

[48]. W. Simpson, *IP in IP Tunneling*, *RFC1853* , October 1995.

[49]. H. Soliman, C. Castelluccia, K. ElMalki and L. Bellier, *Hierarchical Mobile IPv6 (HMIPv6) Mobility Management*, *RFC 5380,* October 2008

[50]. S. Thomson and T. Narten, *IPv6 Stateless Address Autoconfiguration*, *RFC 2462,* 1998.

[51]. S. Thomson, T. Nartenm and T. Jinmei, *IPv6 Stateless Address Autoconfiguration*, *RFC – 4862* , September 2007.

[52]. *US government agencies must be IPv6 ready by July*, Available from: http://www.searchsmbasia.com/en/content/us-goverment-agencies-must-be-ipv6-ready-july [last visited 2010 06/26].

[53]. A.Wei, GouZhi.Wei and Gerard.Dupeyrat, Improving Mobile IPv6 handover and authentication in wireless network with E-HCF, International Journal of Network Management 2009, Wiley InterScience, vol19, issue 6, page: 479-489

[54]. B. Volz, *IETF RFC4649 Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, August 2006.

[55]. Y. Wang, P. Zhang, Y. Zhou , J. Yuan, L. Fang and L. Gen, *Handover Management in Enhanced MIH Framework for HeterogeneousWireless Networks Environment*, WIRELESS PERSONAL COMMUNICATIONS Volume 52, Number 3, 615-636, 2008.

[56]. P. Xavier, T. Marc and H. Hannes, A *Performance Comparison of Mobile IPv6, Hierarchical Mobile IPv6, Fast Handovers for Mobile IPv6 and their Combination*, Mobile Computing and Communications Review, Volume 7, Number 4, 2003.

[57]. *What is IPv6?* http://www.usipv6.com/what_is_ipv6.php [last visited 2010 06/26].

[58]. *Why mobile Japan leads the world* http://www.guardian.co.uk/technology/2007/sep/27/guardianweeklytechnologysection.mobilephones%20Sep%2027,%202007 [last visited 2010 06/26].

[59]. Wikipedia: *Power of Transmitter Output* http://en.wikipedia.org/wiki/Transmitter_power_output, [last visit 2010 06/27]

[60]. Wikipedia: *Sensitivity (electronics)* http://en.wikipedia.org/wiki/Sensitivity_(electronics) , [last visit 2010 06/27]

[61]. Wikipedia: *Log-distance path loss model* http://en.wikipedia.org/wiki/Log-distance_path_loss_model, [last visit 2010 06/27]

[62]. Wikipedia: Noise reduction http://en.wikipedia.org/wiki/Noise_reduction, [last visit 2010 06/27].

Appendix A: Simulation models of MIPv6 and DAD-Less MIPv6 in NS2 and OMNeT++

Most of research publications in networking area report simulation results obtained with a help of by NS2 (Network Simulator 2) [reference]. Therefore, NS was also our original choice in this research project. However, after spending a large amount of time on studying this simulator and trying to acquire the codes for the existing proposals, the simulator has appeared to be inadequate. It is because of the following reasons.

1. The NS2 does not have an IPv6 implementation. Therefore, it is difficult to simulate IPv6 traffic streams for acquiring accurate results of a MIPv6 handover.
2. The most commonly used MIPv6 patch in NS2 is the MobileWan patch which has been developed and released by Motorola in 2001. This patch is originally developed for simulating HMIPv6 in both small and global scale simulation, but the development for the simulation model of large scale has been terminated indefinitely. Since 2001, the patch has not been updated, while the MIPv6 standard had gone through continuous changes till 2004.

For these reasons, much research on MIPv6 conducted with a help of NS2 may have different results from the current standard.

Our next choice was OMNeT++, which is a discrete-event, component-based, modular, open-architecture network simulation environment [reference]. As NS2, it is free and open source for academic research. To perform network simulation, OMNeT++ requires the INET Framework. The INET Framework is an open-source package for simulation of communication networks in the OMNeT++ simulation environment. In the early 2009, an extensible MIPv6 (xMIPv6) simulation model, based on the INET Framework has been released, and it is claimed to be an accurate model of the RFC3775 standard of MIPv6. For these reasons, we have chosen the OMNeT++4.0 and INET Framework for its version 4.0 as tools for our research. The main steps of designing models of MIPv6 and DAD-Less MIPv6 in INET framework are discussed below.

The implementation of DAD-Less MIPv6

The INET Framework generally follows five layers architecture with physical, link, network, transport and application layers; see the circled folders in Figure A.1.

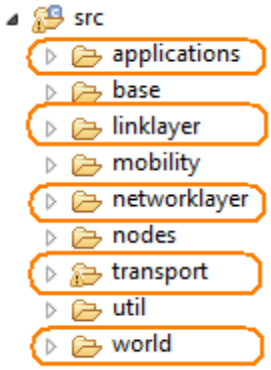

Figure A.1 The source code structure of the INET Framework

As the figure shows, there is no “phyciallayer” folder; this is because all the elements of physical layer are spread under the folders “linklayer” and “world”. The uncircled folders are “base”, “mobility”, “nodes” and “util”. The “base” folder contains the

functions relating to the queues and the sinks. The “mobility” folder contains the functions of mobility models of mobile units which users may define. The “nodes” folder contains the node definitions which can be defined in users’ own demand. At last, the “util” folder contains the functions for accessing the trace files and the XML files.

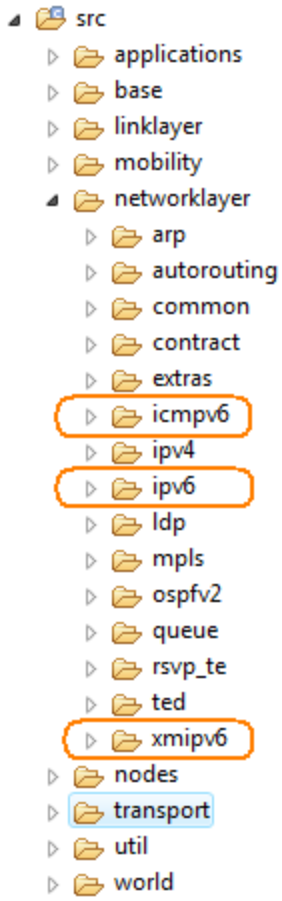

Figure A.2 the components of the xMIPv6

The most important components for the xMIPv6 simulation model are circled in Figure A.2. The behaviour of the Movement Detection process and the Address Configuration process are defined by modifying the classes under the folder “icmpv6”. The files under the “xmipv6” folder mainly serve for the Binding Update process in a

MIPv6 handover. To allow the model to follow the RFC 3775 exactly, the classes under the folder “ipv6” have also been updated. For the details of the MIPv6 handover processes, refer to Chapter 2.

In order to implement a DAD-Less MIPv6 simulation model, one needs to modify the Address Configuration and the Binding Update process of a standard MIPv6 simulation model. Therefore, the classes under the “icmpv6” and the “xmipv6” folders are the main attention when the simulation models are considered.

Appendix B: Simulation configurations file in OMNeT++

```
#
# This ini file runs Telnet sessions on the NClients network, using
# TelnetApp+TCPGenericSrvApp.
#
# See also fileTransfer.ini and basicHTTP.ini for different kinds of
# network traffic.
#

[General]
num-rngs = 2
**.gen[*].rng-0 = 150

debug-on-errors = false

network = mIPv6Network

cmdenv-express-mode = true

tkenv-plugin-path = ../../../Etc/plugins


# number of client computers (MN(s) in our case)
 #NOTE: When increasing the number of MN, make sure that they either have different movement patterns, or different speeds, or (incase they have same speed and movement pattern/direction) then they should not have similar positions on the playground (i.e., their positions should not overlap) or else (for some strange reason) MIPv6 operation will not work
*.total_mn = 1

# number of Servers (CN(s) in our case)
*.total_cn = 2

#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.minIntervalBetweenRAs = 0.03s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 0.07s

#used by the MN to directly access the CN address. This is a parameter in xMIPv6.ned
**.CNAddress = "CN[0]" **.CNAddress1 = "CN[1]"

#**.MN[*].networkLayer.hmipv6Support = false
**.*.networkLayer.hmipv6Support = false

# Setting for DADLess support
#**.*.routingTable6.isDADLess = false
**.*.routingTable6.isDADLess = true



# configurator
*.playgroundSizeX = 850 #channel control
*.playgroundSizeY = 850 #channel control

**.AP*.mobility.x = -1
```

```
**.AP*.mobility.y = -1

# channel physical parameters
*.channelcontrol.carrierFrequency = 2.4GHz
*.channelcontrol.pMax = 2.0mW
#*.channelcontrol.pMax = 100mW
*.channelcontrol.sat = -82dBm
#*.channelcontrol.sat = -82dBm
*.channelcontrol.alpha = 2
#*.channelcontrol.numChannels = 3

# access point

**.MN*.**.mgmt.accessPointAddress = "10:AA:00:00:00:01"
**.wlan.mgmt.numAuthSteps = 4
**.mgmt.frameCapacity = 10

# ALL APs common parameters
**.AP*.wlan.mgmt.beaconInterval = 0.1s

# Access Point AP_Home ; AP_1 ; AP_2 ; AP_3 Parameters for EtherMAC
**.AP_Home.wlan.mgmt.ssid = "HOME"
**.AP_Home.wlan.mac.address = "10:AA:00:00:00:01"
**.AP_Home.eth[0].address = "10:AE:00:00:00:02"
**.AP_Home.eth[0].txrate = 100Mbps
**.AP_Home.eth[0].duplexEnabled = true
**.AP_Home.eth[0].*.scalar-recording = false

**.AP_1.wlan.mgmt.ssid = "AP1"

#the A1:01 specifies AP_1:interface 1
**.AP_1.wlan.mac.address = "10:AA:00:00:A1:01"

#the A1:02 specifies AP_1:interface 2
**.AP_1.eth[0].address = "10:AE:00:00:A1:02"

**.AP_1.eth[0].txrate = 100Mbps
**.AP_1.eth[0].duplexEnabled = true
**.AP_1.eth[0].*.scalar-recording = false


**.AP_2.wlan.mgmt.ssid = "AP2"
#the A2:01 specifies AP_2:interface 1
**.AP_2.wlan.mac.address = "10:AA:00:00:A2:01"

#the A2:02 specifies AP_2:interface 2
**.AP_2.eth[0].address = "10:AE:00:00:A2:02"
**.AP_2.eth[0].txrate = 100Mbps
**.AP_2.eth[0].duplexEnabled = true
**.AP_2.eth[0].*.scalar-recording = false


# mobility
**.MN[0].mobilityType = "RectangleMobility"
**.MN[0].mobility.debug = false
**.MN[0].mobility.x1 = 180
**.MN[0].mobility.y1 = 100
**.MN[0].mobility.x2 = 530
**.MN[0].mobility.y2 = 110
**.MN[0].mobility.startPos = 0
```

```
**.MN[0].mobility.speed = 1mps
**.MN[0].mobility.updateInterval = 0.1s

**.MN[1].mobilityType = "RectangleMobility"
**.MN[1].mobility.debug = false
**.MN[1].mobility.x1 = 170
**.MN[1].mobility.y1 = 100
**.MN[1].mobility.x2 = 530
**.MN[1].mobility.y2 = 110
**.MN[1].mobility.startPos = 0
**.MN[1].mobility.speed = 1mps
**.MN[1].mobility.updateInterval = 0.1s

# udp app setting
**.CN[0].numUdpApps = 1
**.CN[0].udpAppType = "UDPVideoStreamSvr"
**.CN[0].udpApp[*].videoSize = 20MB
**.CN[0].udpApp[*].serverPort = 3088
**.CN[0].udpApp[*].waitInterval = 10ms
**.CN[0].udpApp[*].packetLen = 1000B

**.MN[0].numUdpApps = 1
**.MN[0].udpAppType = "UDPVideoStreamCli"
**.MN[0].udpApp[*].serverAddress = "CN[0]"
**.MN[0].udpApp[*].localPort = 9999
**.MN[0].udpApp[*].serverPort = 3088
**.MN[0].udpApp[*].startTime = 200s

**.CN[1].numUdpApps = 0
**.CN[1].udpAppType = "UDPBasicApp"


# tcp apps setting (off)
#changed from 1 to 0 to turn on tcp apps
**.MN[*].numTcpApps = 0
**.MN[*].tcpAppType = "TelnetApp"

**.MN[0].tcpApp[0].address = "aaaa:b::aaa:ff:fe00:7
**.MN[1].tcpApp[0].address = "aaaa:b::aaa:ff:fe00:8"
**.MN[0].tcpApp[0].port = -1
**.MN[1].tcpApp[0].port = -1
**.MN[*].tcpApp[0].connectAddress = "CN"
**.MN[0].tcpApp[0].connectPort = 1000 #same destination port numbers
**.MN[1].tcpApp[0].connectPort = 1000 #same destination port numbers

**.MN[*].tcpApp[0].startTime = uniform(10,15)
**.MN[*].tcpApp[0].numCommands = exponential(1)
**.MN[*].tcpApp[0].commandLength = exponential(1)
**.MN[*].tcpApp[0].keyPressDelay = exponential(0.1)
**.MN[*].tcpApp[0].commandOutputLength = exponential(40)
**.MN[*].tcpApp[0].thinkTime = truncnormal(2,3)
**.MN[*].tcpApp[0].idleInterval = truncnormal(3600,1200)
**.MN[*].tcpApp[0].reconnectInterval = 30s

#changed from 1 to 0 to turn on tcp apps
**.CN*.numTcpApps = 0
**.CN*.tcpAppType = "TCPGenericSrvApp"
**.CN*.tcpApp[0].address = ""
**.CN*.tcpApp[0].port = 1000
**.CN*.tcpApp[0].replyDelay = 0
```

```
#preceeded all options with .MN[*].pingApp~
# ping app (on)
**.MN[0].pingApp.destAddr = "" #"CN[0]"
**.MN*.pingApp.destAddr = "" #"CN[1]"
**.MN*.pingApp.srcAddr = ""
**.MN*.pingApp.packetSize = 56B
**.MN*.pingApp.interval = 0.01s
**.MN*.pingApp.hopLimit = 32
**.MN*.pingApp.count = 0
**.MN*.pingApp.startTime = 200s                                    #changed from 1
**.MN*.pingApp.stopTime = 0
**.MN*.pingApp.printPing = true

# = ================Added this section for the Correspondent Node (CN) PingApp Parame-
ters=====================
**.CN[0].pingApp.destAddr = "MN[0]"
**.CN[1].pingApp.destAddr = ""
**.CN*.pingApp.destAddr = "" #"MN[0]"
**.CN*.pingApp.srcAddr = ""
**.CN*.pingApp.packetSize = 56B
**.CN*.pingApp.interval = 0.5s
**.CN*.pingApp.hopLimit = 32
**.CN*.pingApp.count = 0
**.CN*.pingApp.startTime = 30s  #10s                                #changed from 1s
**.CN*.pingApp.stopTime = 2000s #450s
**.CN*.pingApp.printPing = true

# =
================================================================================
================================

# tcp settings.
**.tcp.mss = 1024
**.tcp.advertisedWindow = 14336  # 14*mss
**.tcp.sendQueueClass = "TCPMsgBasedSendQueue"
**.tcp.receiveQueueClass = "TCPMsgBasedRcvQueue"
**.tcp.tcpAlgorithmClass = "TCPReno"
**.tcp.recordStats = true

# ip settings
#FIXME
**.routingTableFile = xmldoc("empty.xml")
**.ipv6.procDelay = 10us
**.IPForward = false
**.routingFile = ""

# ARP configuration
**.arp.retryTimeout = 1s
**.arp.retryCount = 3
**.arp.cacheTimeout = 100s
**.networkLayer.proxyARP = true  # Host's is hardwired "false"

# PPP NIC configuration
**.ppp[*].queueType = "DropTailQueue" # in routers
**.ppp[*].queue.frameCapacity = 10  # in routers

# Ethernet NIC configuration
**.eth[*].queueType = "DropTailQueue" # in routers
**.eth[*].queue.frameCapacity = 10  # in routers
```

```
**.eth[*].encap.*.scalar-recording = false
**.eth[*].mac.promiscuous = false
**.eth[*].mac.address = "auto"
**.eth*.mac.txrate = 100Mbps
**.eth*.mac.duplexEnabled = true
**.eth*.mac.*.scalar-recording = false

**.ap.*.scalar-recording = false
**.hub.*.scalar-recording = false

# wireless channels
**.AP_Home.wlan.radio.channelNumber = 1
**.AP_1.wlan.radio.channelNumber = 2
**.AP_2.wlan.radio.channelNumber = 3
**.AP_3.wlan.radio.channelNumber = 4
**.MN*.wlan.radio.channelNumber = 0  # just initially -- it'll scan

# wireless configuration
**.wlan.agent.activeScan = true
**.wlan.agent.channelsToScan = "1 2"  # "" means all
**.wlan.agent.probeDelay = 0.1s
**.wlan.agent.minChannelTime = 0.15s
**.wlan.agent.maxChannelTime = 0.3s
**.wlan.agent.authenticationTimeout = 5s
**.wlan.agent.associationTimeout = 5s

# nic settings
**.mac.address = "auto"
**.mac.maxQueueSize = 14
**.mac.rtsThresholdBytes = 4000B
**.mac.bitrate = 2Mbps
**.wlan.mac.retryLimit = 7
**.wlan.mac.cwMinData = 7
**.wlan.mac.cwMinBroadcast = 31

**.radio.bitrate = 2Mbps
**.radio.transmitterPower = 2.0mW
#**.radio.transmitterPower = 0.1mW
#**.radio.transmitterPower = 100.0mW
**.radio.carrierFrequency = 2.4GHz
**.radio.thermalNoise = -110dBm
**.radio.sensitivity = -82mW
#**.radio.sensitivity = -73mW
**.radio.pathLossAlpha = 2
**.radio.snirThreshold = 4dB
#**.radio.snirThreshold = 2dB

# relay unit configuration
**.relayUnitType = "MACRelayUnitNP"
**.relayUnit.addressTableSize = 100
**.relayUnit.agingTime = 120s
**.relayUnit.bufferSize = 1MB
**.relayUnit.highWatermark = 512KB
**.relayUnit.pauseUnits = 300  # pause for 300*512 bit (19200 byte) time
**.relayUnit.addressTableFile = ""
**.relayUnit.numCPUs = 2
**.relayUnit.processingTime = 2us
**.relayUnit.*.scalar-recording = false

**.debug = true
```

```
**.coreDebug = false

[Config One]
description = "Handover 1_RA-Test1"
sim-time-limit = 250s
repeat = 100
```

Appendix C: Abbreviations

3G	3rd Generation Standards for Mobile Tele-communications
AP	Access Point
AR	Access Router
CLS	Carrying Load Status
CN	Correspondent Node
CoA	Care-of-Address
CTS	Current Tracking Status
DAD	Duplicated Address Detection
DAD-Less MIPv6	Dad-Less Mobile Internet Protocol version 6
DE	Decision Engine
FAR	Foreign Access Router
FLC	Flow Labelling Capability
FMIPv6	Fast-handovers for Mobile Internet Proto-col Version 6
FN	Foreign Network
HA	Home Agent
HD	Handoff Decision
HMIPv6	Hierarchical Mobile Internet Protocol Version 6
HN	Home Network
HN	Handoff Notification

HoA	Home Address
IANA	Internet Assigned Number Authority
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISP(s)	Internet Service Provider(s)
MIPv6	Mobile Internet Protocol version 6
MN	Mobile Node
NAT	Network Address Translation
QoS	Quality of Service
Scast	Simulcast
Soff	Simulcast Off
VoIP	Voice over Internet Protocol
WISP(s)	Wireless Internet Service Provider(s)

Appendix D: Glossary of terms

Access Point	The facility that provides the radio connectivity to MNs.
Access Router	The router that provides Internet connectivity to MNs.
Care-of-Address	A temporary address for an MN while it is not at the HN.
Carrying Load Status	Message contains the information regarding to the loading capacity of the AR.
Correspondent Node	The terminal that is currently communicating with the MN.
Current Tracking Status	Message is used to track the physical location of the MN for DE.
Decision Engine	The entity which uses the information from the CTS and CLS messages to make the handover decision.
Foreign Access Router	Provides Internet connection to an MN except HA. Please note it is not a Foreign Agent as MIPv4. There is no special router in MIPv6.
Foreign Network	The network where the MN is currently connecting with but not HN
Handoff Decision	Message contains the handover decision information from the DE. It informs the entire candidate ARs which AR has been chosen for the MN.
Handoff Notification	Message is sent from the OAR to the MN. It indicates which NAR the MN should connect to.The content of the message is derived from the HD message.
Handover	A process where a mobile device is moved from one

	MIPv6 network to another.
Home Address	A unicast address which is permanently assigned to an MN. Usually the traffic will be delivered to the MN by this HoA directly.
Home Agent	The AR that assigns the HoA to an MN. The assigned HoA should have the same network prefix as the HA.
Home Network	The network where MN has acquired the HoA. It is the network where the HA belongs to.
Mobile Node	A terminal that moves between networks.
Simulcast	Message is the message which starts the Synchronized-Packet-Simulcast (SPS) process. The SPS process is used for separating the packets from the OAR and the MAP. By doing this, there will be less possibility of massing the order of the packets.
Simulcast Off	Message terminates the SPS process.